# AFRL-SN-WP-TR-2002-1055

## RECOGNIZING ARTICULATED OBJECTS IN RANGE IMAGES USING INVARIANTS

**ISAAC WEISS**
**MANJIT RAY**

**UNIVERSITY OF MARYLAND**
**CENTER FOR AUTOMATION RESEARCH**
**COLLEGE PARK, MD 20742**

**FEBRUARY 2002**

**FINAL REPORT FOR PERIOD OF 27 FEBRUARY 1997 – 23 FEBRUARY 2002**

**Approved for public release; distribution unlimited**

**SENSORS DIRECTORATE**
**AIR FORCE RESEARCH LABORATORY**
**AIR FORCE MATERIEL COMMAND**
**WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318**

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 |
|---|---|---|---|



| 1. REPORT DATE (DD-MM-YYYY) | 2. REPORT TYPE | 3. DATES COVERED (FROM - TO) |
|---|---|---|
| 01-02-2002 | Final | 27-01-1997 to 23-02-2002 |

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br>Recognizing Articulated Objects in Range Images Using Invariants<br>Unclassified | 5a. CONTRACT NUMBER<br>F33615-97-1-1015<br>5b. GRANT NUMBER<br>5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S)<br>Weiss, Isaac ;<br>Ray, Manjit ; | 5d. PROJECT NUMBER<br>5e. TASK NUMBER<br>5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME AND ADDRESS<br>University of Maryland<br>Center for Automation Research<br>College Part, MD20742 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS<br>Sensors Directorate<br>Air Force Research Laboratory<br>Air Force Materiel Command<br>Wright-Patterson AFB, OH45433-7318 | 10. SPONSOR/MONITOR'S ACRONYM(S)<br>11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
APUBLIC RELEASE
,

13. SUPPLEMENTARY NOTES

14. ABSTRACT
See report

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| | | | Same as Report (SAR) | 73 | EM107, (blank)<br>lfenster@dtic.mil |
| a. REPORT<br>Unclassified | b. ABSTRACT<br>Unclassified | c. THIS PAGE<br>Unclassified | | | 19b. TELEPHONE NUMBER<br>International Area Code<br>Area Code Telephone Number<br>703767-9007<br>DSN<br>427-9007 |

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39.18

# NOTICE



THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Greg Arnold, Ph.D.
Project Engineer
ATR & Fusion Algorithms Branch

DALE E. NELSON, Ph.D.
Chief, ATR & Fusion Algorithms Branch
Sensors Directorate

DAVE CHANDLER, LtC, USAF
Deputy Chief
Sensor ATR Technology Division
Air Force Research Laboratory
Wright-Patterson AFB, Ohio 45433

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| **1. REPORT DATE** ***(DD-MM-YY)*** | **2. REPORT TYPE** | **3. DATES COVERED** ***(From - To)*** |
|---|---|---|
| February 2002 | Final | 02/27/1997 – 02/23/2002 |

**4. TITLE AND SUBTITLE**
RECOGNIZING ARTICULATED OBJECTS IN RANGE IMAGES USING INVARIANTS

**5a. CONTRACT NUMBER** F33615-97-1-1015
**5b. GRANT NUMBER**
**5c. PROGRAM ELEMENT NUMBER** 62301E

**6. AUTHOR(S)**
ISAAC WEISS
MANJIT RAY

**5d. PROJECT NUMBER** ARPA
**5e. TASK NUMBER** AA
**5f. WORK UNIT NUMBER** 1Q

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
UNIVERSITY OF MARYLAND
CENTER FOR AUTOMATION RESEARCH
COLLEGE PARK, MD 20742

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318

**10. SPONSORING/MONITORING AGENCY ACRONYM(S)**
AFRL/SNAT

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)**
AFRL-SN-WP-TR-2002-1055

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited.

**13. SUPPLEMENTARY NOTES**
Report contains color.

**14. ABSTRACT** ***(Maximum 200 Words)***

Articulated targets such as tanks can have many degrees of freedom, in addition to the unknown variables of viewpoint. Recognizing such a target in an image can involve a search in a high-dimensional space that involves all these unknown variables. In this project we use invariance to reduce this search space to a manageable size. Our method avoids feature detection for improved robustness. This is done by dealing with large parts of the visible object as wholes rather than with individual features. More specifically, we define a grid on the image, and draw a sphere around each grid point. We fit a 3D quadric to the object part contained within the sphere. We then find the Euclidean invariants of the quadric and assign them to the grid point, thus obtaining an invariant representation of the image. This is done at various scales by changing the sphere radii and grid spacing. The invariant representation is than matched against ones obtained from known models at various articulations. We apply the method to range images of objects such as backhoes.

**15. SUBJECT TERMS**
Automatic Target Recognition, Invariance, Laser Radar, Obscuration, Articulation Recognition by Parts

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT:** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** (Monitor) |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | Greg Arnold |
| Unclassified | Unclassified | Unclassified | SAR | 78 | **19b. TELEPHONE NUMBER** *(Include Area Code)* (937) 255-1115 x4388 |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std. Z39-18

# Contents

## List of Figures

# Preface

| | |
|---|---|
| Contractor name: | DARPA/AFRL |
| Contrator address: | AFRL/SNAT, Bldg 620<br>2241 Avionics Circle<br>WPAFB, OH 45433-7321 |
| Contract number: | F33615-97-1-1015, ARPA Order E655 |
| Inclusion dates: | Feb. 1997 to Feb. 2002 |
| Contract Manager: | Vince Velten, Greg Arnold |
| Office symbol: | AFRL/SNAT |

The authors with to thank Vince Velten and Greg Arnold of the AFRL Wright Laboratory, and T.J. Klausutis of AFRL at Eglin, for their very active monitoring of the research and many useful scientific discussions, as well as taking care of the administative side.

# 1

# Summary

Model-based object recognition from single images constitutes a prominent area of research in computer vision. In a typical task, one has appropriate representations of a set of objects in the form of models or reference images, and the objective is to determine whether an object from this set appears in an image taken from an unknown viewpoint with a camera of unknown intrinsic geometry.

The simplest approach toward achieving this goal is to use methods that are variants of template matching, whereby every possible transformation of a reference image is compared with the test image in order to verify a match. A reduction in the search effort can be achieved either by appropriately sampling the search space or by partitioning it into regions where the aspect of the object remains essentially unchanged. Recognition can also be achieved using a smaller set of templates derived from the original set using methods that take advantage of the fact that reference images from neighboring viewpoints are highly correlated.

Alternatively, the search space can be significantly reduced through the use of invariants. These are functions of feature configurations that remain unchanged under the action of different transformation groups. These functions should be distinctive enough so that the number of model feature configurations that can match a given configuration of image features is reasonably small.

The principal motivation behind this work is to develop a methodology for recognizing objects that does not depend on the detection of discrete features in the image, since it might be difficult to do so for noisy data. The images considered are range images since they convey more information about the scene by incorporating the depths of the visible points. The schemes described here use scale-space invariants with the canonical frame paradigm to achieve some degree of independence to changes in viewpoint, resolution and the presence of occlusions.

# 2

# Introduction

An important aspect of any recognition system is the type of sensor it employs to collect data. The sensor determines the resolution (the total number and frequency of sample points) and precision (the precision of each sample point). More importantly, it determines whether the data provides 2D or 3D information about the scene. Schemes for acquiring 3D information from a scene can be classified as taking one of two approaches: passive or active. In the passive approach, 3D information is inferred from the scene using existing energy in the environment, such as reflected light. In the active approach, the 3D information is derived by projecting energy, such as sonar waves or laser light. 3D data recovered from passive approaches generally lack the necessary precision and resolution for 3D object recognition; hence in this work we will focus on 3D data obtained from active sensors, commonly referred to as *range data* or *range images*.
Active range sensing can be divided into two main classes:

1. In the first class, the principle of triangulation is used. A point in the scene is illuminated and observed by a sensor. Then, since the distance between the light source and the camera (referred to as the baseline) is known together with the angles of the original and reflected rays, the depth can be determined by the principles of triangulation. A disadvantage of this method is the presence of *shadow areas* which are areas of the scene for which no data point is visible since they are not visible both from the light source and the camera.

2. In the second class of sensors, known as time-of-flight sensors, a signal is emitted, its return time is measured and used in calculating the depth. This eliminates the presence of shadow regions. This can be done using either a pulsed laser, where discrete time intervals are measured, or a continuous-beam laser where the phase shift of the reflected signal with respect to the original is used to measure the depth.

Once the necessary measurements from the scene have been made, the data must be represented using symbolic descriptions to enable the system to carry out high-level recognition processes. In most work, the first step in obtaining this symbolic representation is the partitioning of the input based on the desired description. Range scanners sample points on surfaces, hence the vision system must rely on the information derived from surfaces. A useful survey of the methods used in deriving such a description from range images can be found in [Reference 1].
A number of works have focused on the segmentation of range images. In range images, local surface properties, such as surface normal and surface curvature, have been used to achieve segmentation. Besl and Jain [Reference 3] use Gaussian and mean curvature sign labels to achieve a coarse segmentation. For each of the segmented regions, a *seed* region is selected and an approximate surface is fitted to it. The seed regions then serve to grow the final segmented regions. Yang [Reference 26] uses a pyramid to achieve a similar segmentation at different scales. Hoffman and Jain [Reference 8] segment and classify regions into connected planar, convex and concave surfaces using various statistical measurements. A common aspect of these approaches is that pixels with similar properties are grouped together to form the desired regions. A

second approach to segmentation is to detect dissimilarities among data points and to partition them accordingly. Using the outputs of three segmentation methods, Fan et al. [Reference 5] detect discontinuities in the curvature of the input data to partition range images.

Once the scene has been segmented into regions, it is necessary to extract a description that can be employed in matching. Ideally, such a description would be unambiguous, unique, not sensitive to the absence of some data points as in the case of occlusion, and convenient to store and use in matching. Representations for 3D object descriptions can be generally classified into:

1. *Surface-based representations*: In this class of representations, surface properties such as surface normals and Gaussian curvatures are approximated and used in the description of the collected data points. In [Reference 9], the orientation of the surface normal at each surface point is mapped onto the corresponding point on the unit sphere (*Gaussian sphere*). Several modifications exist; one that has particular relevance to this work is to assign a weight to each point equivalent to the corresponding surface patch area on the object [Reference 9]. Weiss [Reference 22] uses a representation that is viewpoint independent, namely it is invariant under projective transformations.

2. *Discontinuity-based representations*: Rather than storing information about surfaces, these representations preserve information about the points where the surface characteristics change. The results are curves embedded in 3D space and can be represented compactly using mathematical formulations. Mokhtarian [Reference 13] uses the general expressions for the curvature and torsion of space curves to describe these curves over several scales. In [Reference 15], several types of discontinuities are detected and modeled as *steps* (where the depth map is discontinuous), *roofs* (where the surface normal is discontinuous), *smooth joins* (where the principal curvature is discontinuous), and *shoulders* (which are combinations of two roofs). Weiss and Ray [Reference 24] represent quintuples of 3D points as lines in a 3D invariant space. *Aspect graphs* [Reference 11] have been used to represent rigid polyhedral objects [Reference 16] and general curved objects [Reference 18]. Each node in an aspect graph represents a distinct 2D viewpoint of a 3D object and the edges represent transformations or *visual events*. Essentially, the graph partitions the viewpoint directions into stable regions, i.e. regions in which small changes in the viewpoint directions do not change the aspect of the object. Godin and Levine [Reference 6] use *discontinuity labeling* to construct an edge-junction graph using crease and jump edges.

3. *Volumetric representations*: These representations represent volumes rather than surfaces or their discontinuities. *Superquadrics* [Reference 2] are an extension of basic quadric surfaces and solids and several parameters are used to determine the object's squareness, size, bending, tapering and twisting. Non-linear minimization techniques are used to recover these parameters from a set of data [Reference 17]. *Octrees* are a second class of volumetric representation, constructed by recursively decomposing a cubical volume until each resultant subcube is homogeneous with respect to some criterion [Reference 10]. In *sweep representations*, the desired 3D shape is described by the sweeping action of a 2D element. The sweeping path is referred to as the *spine*, the element swept as the *cross section*, and the geometrical relationships of the cross section and the spine as the sweeping rule. The most widely used representations of this type are *generalized cylinders* [Reference 4].

# 3

# Overview

The principal objective of this work is to develop a framework whereby we can recognize objects in a single range image under the following constraints:

1. The segmentation of range images into homogeneous regions is vulnerable not only to errors in the data associated with each image point but also to the inherent difficulties of the segmentation process. Consequently, it is desirable to develop a system that is not dependent on the extraction of specific features or the estimation of a description based on grouping features that share certain characteristics. The resultant system will tend to have a greater tolerance to image noise.

2. It is possible that the object in the range image that is targeted for recognition is occluded by other objects in the scene. Consequently, not only is the data pertaining to certain components of the object missing, but an added difficulty has to be overcome in that the occluding clutter needs to be segmented out. It would be helpful if there were a mechanism that is more resistant to such occlusion.

3. An additional problem that will be targeted is the question of articulated objects. Articulation is the displacement of one component of the object with respect to another. Such deformations are not random but can be parameterized. Formulating a matching algorithm for articulated objects often involves a search in a high-dimensional space that incorporates all possible degrees of freedom of the object. We will attempt to formulate a representation for the space that will enable us to carry out this search efficiently.

An important characteristic of range images that makes recognition under these constraints feasible is that the depth is known at each point of the image. Since the descriptions of the object models are also three-dimensional, there is no loss in dimensionality. Also since the transformation between the model and the range image is a Euclidean transformation, it is simple to extract a large number of measurables that are invariant to such transformations.

However, most classic invariants suffer from one of two problems that makes it difficult to use them in the present context. Most plane invariants, Euclidean or otherwise, can be classified into two broad categories:

- *Global invariants*: Such invariants are derived from the data points associated with the entire object. For instance, moment invariants are used for describing shape or fitting planes or surfaces to homogeneous segmented regions. However, while these invariants are relatively insensitive to the presence of image noise, they are vulnerable to occlusion when data pertaining to portions of the object are missing.

- *Local invariants*: These invariants are extracted at each point of the range image from data associated with the points in the neighborhood of each point. Examples include surface normals and curvature, which are both relative invariants. Since these invariants are dependent only on local neighborhoods of points, they are relatively insensitive to occlusion. However, since these invariants generally involve differentials or differences, they require that the points be well localized.

It is, however, possible to derive invariants that serve as a compromise between these two extremes.

### 3.1 Scale-space invariants

Scale-space invariants can best be described as global invariants of local parts. They are functions which take as parameters not only the coordinates $(x, y, z)$ of a point on the range image but also a scale parameter $r$, and are defined in terms of some global invariant characteristics of all the points of the range image lying within distance $r$ of the point $(x, y, z)$. Thus, if the global invariant function of interest is the parameters of the surface fitted to the neighborhood of a point, then the scale-space equivalent would the surface parameters fitted to neighborhoods of increasing radii.

Ideally, we would like to compute such scale-space invariants for each point of the range image and for all possible values of the scale parameter. However, in practice, it is necessary to calculate these functions only at discrete values of the geometric and scale parameters. This discretization can be done either uniformly, as is usually done in the case of the geometric parameters since segmentation of the range data has to be avoided, or only at positions where the invariant function shows significant changes, as can be done in the case of the scale parameter (since if they are suitably designed, the global invariant functions can be made to change significantly only if significant new artifacts are introduced into the neighborhood by increasing the scale parameter). However, in this work, both the geometric and scale parameters are discretized uniformly.

For each of the discrete neighborhoods so determined, we need to determine functions that are Euclidean invariants and are dependent on the data present in the entire neighborhood. It is possible to calculate moments of this data and then derive absolute invariants by taking appropriate combinations of individual moment functions, but in the calculations of these functions, the data points lying at the boundaries of each neighborhood contribute the most. Thus, if data points are added to the boundaries by increasing the scale parameter, the values of the functions will undergo significant changes. Hence, we use the following approach to derive invariant functions:

1. In order to assign more natural weights to the data points in a neighborhood, we fit a quadratic surface to all the points. It is necessary to determine these parameters in such a manner that they are invariant to the particular class of transformations we are interested in, namely Euclidean transformations.

2. Once the surface parameters have been determined, we can employ the canonical frame paradigm to generate a canonical frame dependent only on these parameters and map all the data points (or some of them after selecting an appropriate uniform grid) to the canonical frame.

3. If the surface parameters correspond to those of a regular conic, then the canonical frame can be easily determined by mapping the center of the conic to the origin and its principal axes to the coordinate axes of the frame. The exact ordering of mapping the axes can be based on heuristics; for instance, we can map the principal axes in the order of increasing length.

### 3.2 Recognition

The entire recognition process can be summarized as follows:

1. Each model is represented by invariant characteristics derived from a number of range images obtained from different viewpoints. Images from two different viewpoints are required only when considerably different portions of the model are visible in the two.

2. Each such range image of a model is sampled using a grid of fixed resolution. The resolution of this grid is unimportant as long as it is reasonably dense.

3. Each data point in the sampled range image is associated with some invariant characteristics for each of a specified set of values for the scale parameter. These characteristics form the basis for the recognition process.

4. Given a test range image in which the object has to be identified, the object is first isolated by removing the background in its neighborhood. This is done under the assumption that the background is essentially a plane and occupies a larger portion of the range image than any single plane. This isolation does not need to be perfect, but is essential in order to prune a large amount of irrelevant data.

5. Once the object has been isolated, the data points are again sampled using a regular grid of some resolution. This sampling resolution does not need to match the one at which the model data points are sampled; it can be much sparser.

6. Invariant characteristics are generated for each of the sampled points in a manner identical to that used for the model points. These are then compared to the sets of model invariant points and points that are sufficiently close are considered as initial matches.

7. The initial matches so obtained are purged to retain only those in which the spatial relationships between the image data points are consistent with those between the model data points.

8. Scores are assigned to each model depending on the number of initial matches associated with each and the closeness of the model invariant points to the points derived from the test range image. Only a few models with the best scores are kept, all others are discarded.

9. For each probable match retained, the Euclidean transformation between the image and model data points is calculated and the model range image is transformed into the scope of the test range image so that a per-pixel comparison is possible. The model for which such a comparison yields the greatest correlation is deemed to correspond to the object.

# 4

# Invariant Descriptors

Invariant descriptors of any geometric structure can be obtained by deriving a canonical frame of reference dependent only on the configuration of the structure and transforming different geometric properties of the structure to this reference frame. Specifically, in our scheme, a quadratic surface is fitted to the points in the neighborhood of a data point, the parameters of which are then used to define a canonical frame. The coordinates of the data point can then be transformed to this frame. These transformed coordinates are the basic invariant descriptors used here. The specific issues that are addressed in the next sections are:

1. What is the nature of the neighborhood used to obtain the surface?
2. Why is the neighborhood of a data point that is not necessarily a quadratic surface represented as one? Note that in this context we are not talking about local neighborhoods but about extended ones, depending on the value of the scale parameter.
3. How are the parameters of the surface derived so that they are invariant to Euclidean transformations?
4. How is the canonical frame obtained from the parameters of the surface?

# 5

# Neighborhoods

The neighborhood of a data point is defined as all the data points of the range image that lie within a sphere of some specified radius centered at the particular data point. There exists a close relationship between the parameters of the quadratic surface fitted to a neighborhood and its radius or extent since increasing its extent brings new components of the object within the neighborhood resulting in a corresponding change in the surface parameters.

This does not imply that we have to closely sample the entire range of possible radii and store invariant descriptors for the data point at each sample. It is possible to find ranges of radius values where no new components are introduced into the neighborhood and so the invariant descriptors of the data point do not undergo significant change.

However, a different approach is used in this work. Instead of trying to determine ranges of radius values for each model range image, we sample the total range of radius values coarsely and use the *same* set of ranges while mapping the points derived both from the model and the test range images to the canonical frame.

# 6

# The Quadratic Surface

The quadratic surface is the lowest order 3D geometric structure from which properties invariant to Euclidean transformations can be extracted. Even though higher-order structures would yield more invariant properties, fitting such structures to data points sampled from a very different underlying structure would result in parameter values that are unreliable. Moreover, with the exception of the degenerate case when the surface is exactly a plane, enough information can be extracted from the parameters of such a surface to obtain a 3D canonical frame.

A quadratic surface is a second-order algebraic surface given by the general equation

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0 \tag{1}$$

Define the following:

$$e = \begin{pmatrix} a & h & g \\ h & b & f \\ g & f & c \end{pmatrix} \tag{2}$$

$$E = \begin{pmatrix} a & h & g & p \\ h & b & f & q \\ g & f & c & r \\ p & q & r & d \end{pmatrix} \tag{3}$$

The quadratic surface can then be represented as $X^t E X = 0$ where $X = (x, y, z, 1)^t$. Quadratic surfaces are also called *quadrics*; there are 17 standard-form types depending on the rank and eigenvalues of $e$ and the rank and determinant of $E$. For instance, if the rank of $e$ is 1, then the surface represents either coincident or parallel planes depending on whether the rank of $E$ is 1 or 2 respectively. On the other hand, if the ranks of $e$ and $E$ are 3 and 4 respectively, then the surface represents either an ellipsoid (if all the eigenvalues of $e$ have the same sign) or a hyperboloid (otherwise).

The parameters of this surface have to be determined in such a manner that they fulfill the following requirements:

1. They are independent of the viewpoint from which the range data is obtained.

2. They are invariant to the size of the object in the image and the pixel resolution of the image.

The following sections describe how each of these requirements is addressed.

# 7

# Viewpoint Invariance

Ideally, in order to fit a quadratic surface to a set of 3D points (which may be scattered and not suitably modeled by a quadratic surface) in a manner that is invariant under rotations and translations, we must minimize a sum of squared errors of the data points, measured as the distances of the data points from the fitted surface in a direction normal to the surface. Determining the surface parameters using this measure is difficult and requires non-linear least squares minimization. However, since this minimization often yields a local minimum rather than the global one, especially when the initial estimate is not close the global minimum, it is necessary to investigate other measures that can be used to determine the surface parameters while still fulfilling the invariance requirements.

## 7.1 Approximate distance measure

This measure was proposed in [Reference 21], where the problem of parametric representations and estimations of complex curves in 2D, surfaces in 3D and non-planar space curves in 3D is addressed. Curves and surfaces can be represented either parametrically or implicitly, and the latter representation is chosen here.

Let $f : \Re^n \to \Re^k$ be a *smooth* map, a map with continuous first and second derivatives at every point. The set $Z(f) = \{x : f(x) = 0\}, x \in \Re^n$ of zeros of $f$ is defined by the *implicit equations*

$$f_1(x) = 0, \ldots, f_k(x) = 0 \tag{4}$$

For instance, $Z(f)$ is a planar curve if $n = 2$ and $k = 1$, a surface if $n = 3$ and $k = 1$, and a space curve if $n = 3$ and $k = 2$. In general, the distance of a point $x \in \Re^n$ to the set of zeros $Z(f)$ cannot be found directly.

The case of a linear map is an exception, in which the Jacobian matrix $\mathbf{D} = Df(x)$ is constant, and we have the identity

$$f(y) \equiv f(x) + \mathbf{D}(y - x) \tag{5}$$

Without loss of generality, we will assume that the rank of $\mathbf{D}$ is $k$. The unique point $\hat{y}$ that minimizes the distance $||y - x||$ to $x$, constrained by $f(y) = 0$, is given by

$$\hat{y} = x - \mathbf{D}^{\dagger} f(x) \tag{6}$$

where $\mathbf{D}^{\dagger}$ is the pseudo-inverse of $\mathbf{D}$ which in this case is $\mathbf{D}^{\dagger} = \mathbf{D}^t(\mathbf{D}\mathbf{D}^t)^{-1}$. The square of the distance from $x$ to $Z(f)$ is

$$\mathrm{dist}(x, Zf)^2 = ||\hat{y} - x||^2 = f(x)^t(\mathbf{D}\mathbf{D}^t)^{-1} f(x) \tag{7}$$

In the non-linear case, we approximate the distance from $x$ to $Z(f)$ with the distance from $x$ to the set of zeros of a *linear model* of $f$ at $x$, which is a linear map $\hat{f} : \Re^n \to \Re^k$ such that

$$f(y) - \hat{f}(y) = O(||y - x||^2) \tag{8}$$

and is given by the truncated Taylor series expansion of $f$

$$\hat{f}(y) = f(x) + Df(x)(y - x) \tag{9}$$

Clearly $\hat{f}(x) = f(x)$, $D\hat{f}(x) = Df(x)$ and we have

$$\mathrm{dist}(x, Zf)^2 \approx f(x)^t (Df(x) Df(x)^t)^{-1} f(x) \tag{10}$$

which we call the *approximate distance* from $x$ to $Z(f)$. For $k = 1$, which is the case for surfaces, $Df(x) = \nabla f(x)^t$ and the right hand side of Equation 10 reduces to $f(x)^2 / \|\nabla f(x)\|^2$.

## 7.2 Properties

The approximate distance measure has the following geometric properties:

- It is independent of the representation of $Z(f)$. If $g(x) = Af(x)$ for a non-singular $k \times k$ matrix $A$, then

$$\begin{aligned} g(x)^t (Dg(x) Dg(x)^t)^{-1} g(x) &= f(x)^t A^t (A Df(x) Df(x)^t A^t)^{-1} A f(x) && (11) \\ &= f(x)^t (Df(x) Df(x)^t)^{-1} f(x) && (12) \end{aligned}$$

- It is invariant to Euclidean transformations. If $Rx + t$ is the transformation, then $D(f(Rx + t)) = Df(Rx + t)R$ and therefore

$$\begin{aligned} Df(Rx + t) Df(Rx + t)^t &= Df(Rx + t) R R^t Df(Rx + t)^t && (13) \\ &= Df(Rx + t) Df(Rx + t)^t && (14) \end{aligned}$$

## 7.3 Surface fitting

Since we are interested in fitting curves and surfaces to data, we restrict ourselves to maps described by a set of parameters. Thus, we only consider maps $f : \Re^n \to \Re^k$ which can be written as

$$f(x) \equiv \phi_\alpha(x) \tag{15}$$

for the parameters $\alpha = \{\alpha_1, \ldots, \alpha_r\}^t$. The approximate distance of $x$ to $Z(f)$ can be denoted by

$$\delta(\alpha, x) = \sqrt{\phi_\alpha(x)^t (D\phi_\alpha(x) D\phi_\alpha(x)^t)^{-1} \phi_\alpha(x)} \tag{16}$$

The *approximate mean square distance* for a set of points $\mathcal{D} = \{p_1, \ldots, p_q\}$ is given by

$$\Delta^2_{\mathcal{D}}(\alpha) = \frac{1}{q} \sum_{i=1}^{q} \delta(\alpha, p_i)^2 \tag{17}$$

Depending on the particular parameterization $\phi$, the $r$ parameters $\alpha_1, \ldots, \alpha_r$ may not be independent. For example, by Equation 12, the sum does not change when we replace $f$ by $Af$, and if $\hat{f}$ minimizes Equation 17, so does $A\hat{f}$. In other words, the parameters may not be *identifiable*. Since we are not interested in $\hat{f}$ but in $Z(\hat{f})$, we can counter this by imposing the symmetric matrix constraint

$$\frac{1}{q} \sum_{i=1}^{q} Df(p_i) Df(p_i)^t = I_k \tag{18}$$

which is equivalent to $\frac{k(k+1)}{2}$ scalar constraints on the parameters.
The solution of Equation 17 subject to Equation 18 can be done using such non-linear least squares minimization techniques as the Levenburg-Marquardt algorithm.
However, if we make the further assumption that the matrix function $Df(x)Df(x)^t$ is *constant* on $Z(f)$, (for $k = 1$, this means that the length of the gradient is constant on $Z(f)$, which is true for a sphere), and the points $\{p_i\}_{i=1}^{q}$ are close to $Z(f)$, we have

$$I_k = \frac{1}{q}\sum_{i=1}^{q} Df(p_i)Df(p_i)^t \approx Df(p_j)Df(p_j)^t \tag{19}$$

for some $j$ by continuity of $Df$. We can then replace the approximate mean square error $\Delta_D^2(\alpha)$ by the *mean square error*

$$\xi_{\mathcal{D}}^2(\alpha) = \frac{1}{q}\sum_{i=1}^{q} ||f(p_i)^2|| \tag{20}$$

The *global minimum* of Equation 20 subject to Equation 19 can be found linearly by using a generalized eigenvalue fit, as described next.

## 7.4 Generalized eigenvalue fit

Let $X_1(x), \ldots, X_h(x)$ be polynomials and denote

$$X = (X_1, \ldots, X_h)^t : \Re^n \to \Re^h \tag{21}$$

For instance, $X$ could be $(1, x_1, x_2, x_1^2, x_1x_2, x_2^2)^t$.
Let the map be represented as $f = FX : \Re^n \to \Re^k$ for a $k \times h$ matrix $F$ of parameters which have to be determined. Since differentiation is a linear operator

$$Df = D[FX] = F[DX] \tag{22}$$

and the constraint in Equation 19 is given by

$$I_k = \frac{1}{q}\sum_{i=1}^{q} F[DX(p_i)][DX(p_i)^t]F^t = FN_{\mathcal{D}}F^t \tag{23}$$

where

$$N_{\mathcal{D}} = \frac{1}{q}\sum_{i=1}^{q}[DX(p_i)DX(p_i)^t] \tag{24}$$

and is symmetric nonnegative definite. The mean square error (Equation 20) is given by

$$\begin{aligned}
\xi_{\mathcal{D}}^2(\alpha) &= \frac{1}{q}\sum_{i=1}^{q} ||FX(p_i)||^2 && (25)\\
&= \frac{1}{q}\sum_{i=1}^{q} \mathrm{trace}(F[X(p_i)X(p_i)^t]F^t) && (26)\\
&= \frac{1}{q}\mathrm{trace}(FM_{\mathcal{D}}F^t) && (27)
\end{aligned}$$

where

$$M_{\mathcal{D}} = \frac{1}{q}\sum_{i=1}^{q}[X(p_i)X(p_i)^t] \tag{28}$$

which is the covariance matrix of $X$ over the data set $\mathcal{D}$.

It is shown in [Reference 21] that the minimizer $\hat{F} = \{\hat{F}_i\}^k_{i\pm}$ of Equation 27 subject to Equation 23 is given by the eigenvectors corresponding to the $k$ smallest eigenvalues $0 \leq \lambda_1 \leq \cdots \lambda_k$ of the linear system

$$\hat{F}_i M_{\mathcal{D}} = \lambda_i \hat{F}_i N_{\mathcal{D}} \tag{29}$$

## 7.5 Invariance to Euclidean transformations

If $T(x) = Rx + t$ is an Euclidean transformation, then the polynomial vector $X$ will be transformed as

$$X(T(x)) = T^* X(x) \tag{30}$$

where it is shown in [Reference 21] that $T^*$ satisfies $(T^{-1})^* = (T^*)^{-1}$. For instance, if

$$T(x) = \begin{pmatrix} x_1 + b_1 \\ x_2 + b_2 \end{pmatrix} \tag{31}$$

and $X$ is the vector of monomials of degree $\leq 2$ in two variables $X = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$ then

$$\begin{aligned} X(T(x)) &= \begin{pmatrix} 1 \\ x_1 + b_1 \\ x_2 + b_2 \\ (x_1 + b_1)^2 \\ (x_1 + b_1)(x_2 + b_2) \\ (x_2 + b_2)^2 \end{pmatrix} && (32) \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ b_1 & 1 & 0 & 0 & 0 & 0 \\ b_2 & 0 & 1 & 0 & 0 & 0 \\ b_1^2 & 2b_1 & 0 & 1 & 0 & 0 \\ b_1 b_2 & b_2 & b_1 & 0 & 1 & 0 \\ b_2^2 & 0 & 2b_2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix} && (33) \\ &= T^* X(x) && (34) \end{aligned}$$

If $f = FX$ is a solution of the generalized eigenvalue fit for the data set $\mathcal{D}$ and $T$ is a Euclidean transformation, then $g = f \circ T = (FT^*)X$ is the solution for the same problem for the data set $T^{-1}[\mathcal{D}]$ because

$$\begin{aligned} M_{T^{-1}[\mathcal{D}]} &= \frac{1}{q} \sum_{i=1}^{q} X(T^{-1}(p_i)) X(T^{-1}(p_i))^t && (35) \\ &= \frac{1}{q} \sum_{i=1}^{q} ((T^{-1})^* X(p_i))((T^{-1})^* X(p_i))^t && (36) \\ &= (T^*)^{-1} M_{\mathcal{D}} (T^*)^{-1} && (37) \end{aligned}$$

which implies that

$$(FT^*) M_{T^{-1}[\mathcal{D}]} (FT^*)^t = F M_{\mathcal{D}} F^t \tag{38}$$

Similarly, for $N_{\mathcal{D}}$ we have, by the chain rule,

$$D(X(T(x))) = (DX)(T(x)) \cdot D(T(x)) = T^* DXR \tag{39}$$

which implies, since $R$ is orthonormal, that

$$N_{T^{-1}[\mathcal{D}]} = (T^*)^{-1} N_{\mathcal{D}} (T^*)^{-1} \tag{40}$$

and so

$$(FT^*) N_{T^{-1}[\mathcal{D}]} (FT^*)^t = F N_{\mathcal{D}} F^t \tag{41}$$

Thus, the solution obtained by the generalized eigenvalue fit will be invariant to $T$.

### 7.6 Scheme for a quadratic surface

Recall that a quadratic surface is a second-order algebraic surface of the form

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0 \tag{42}$$

Thus, the values of the parameters $\alpha = (a, b, c, f, g, h, p, q, r, d)^t$ can be estimated from a set of data points $\mathcal{D} = \{p_1, \ldots, p_q\}$, where each $p_i$ has coordinates $(x_i, y_i, z_i)$, as follows:

1. Since $M_{\mathcal{D}}$ is of the form

$$M_{\mathcal{D}} = \frac{1}{q} \sum_{i=1}^{q} [X(p_i) X(p_i)^t] \tag{43}$$

it can be calculated for a quadratic surface as follows:

(a) For each point $p_i$, form $c_i$ and $cov_i$ as

$$\begin{aligned} c_i &= \begin{pmatrix} x_i^2 & y_i^2 & z_i^2 & 2y_iz_i & 2z_ix_i & 2x_iy_i & 2x_i & 2y_i & 2z_i & 1 \end{pmatrix}^t && (44) \\ cov_i &= c_i c_i^t && (45) \end{aligned}$$

(b) $M_{\mathcal{D}}$ is thus given by

$$M_{\mathcal{D}} = \frac{1}{q} \sum_{i=1}^{q} cov_i \tag{46}$$

2. Since $N_{\mathcal{D}}$ is of the form

$$N_{\mathcal{D}} = \frac{1}{q} \sum_{i=1}^{q} [DX(p_i) DX(p_i)^t] \tag{47}$$

it can be calculated as follows:

(a) For each point $p_i$, form $\frac{\partial c_i}{\partial x}$, $\frac{\partial c_i}{\partial y}$, $\frac{\partial c_i}{\partial z}$, and $jcov_i$ as

$$\begin{aligned} \frac{\partial c_i}{\partial x} &= \begin{pmatrix} 2x_i & 0 & 0 & 0 & 2z_i & 2y_i & 2 & 0 & 0 & 0 \end{pmatrix}^t && (48) \\ \frac{\partial c_i}{\partial y} &= \begin{pmatrix} 0 & 2y_i & 0 & 2z_i & 0 & 2x_i & 0 & 2 & 0 & 0 \end{pmatrix}^t && (49) \\ \frac{\partial c_i}{\partial z} &= \begin{pmatrix} 0 & 0 & 2z_i & 2y_i & 2x_i & 0 & 0 & 0 & 2 & 0 \end{pmatrix}^t && (50) \\ jcov_i &= \frac{\partial c_i}{\partial x} \frac{\partial c_i}{\partial x}^t + \frac{\partial c_i}{\partial y} \frac{\partial c_i}{\partial y}^t + \frac{\partial c_i}{\partial z} \frac{\partial c_i}{\partial z}^t && (51) \end{aligned}$$

(b) $N_{\mathcal{D}}$ is thus given by

$$N_{\mathcal{D}} = \frac{1}{q} \sum_{i=1}^{q} jcov_i \tag{52}$$

3. $\alpha$ can be determined as the eigenvector corresponding to the smallest eigenvalue of the generalized eigenvalue problem

$$M_{\mathcal{D}}x = \lambda N_{\mathcal{D}}x \tag{53}$$

which can be solved using the QZ factorization algorithm found in [Reference 14]. Note that in the solution so obtained, $\alpha$ is normalized such that $||\alpha|| = 1$.

# 8

# Resolution Invariance

This section addresses the problem of attaining invariance to changes in resolution of the range image. Differences in resolution of the components of the objects can be the result of any of the following:

1. Since the range image is actually a sampling of the visible surface of the object using an uniform grid of some resolution, changes in the grid resolution will result in a corresponding change in the resolution of the image. This is what is commonly termed a change in image resolution.

2. A more subtle change occurs when the viewpoint varies. As the viewpoint changes, the contributions of different portions of the visible components of the object to the range image change. This also causes a corresponding change in the contribution of each such portion to the mean square error function being minimized in the generalized eigenvalue fit described in Section 7.4, since the number of data points derived from each portion changes. This will cause the minimization process to produce a surface that is closer to those portions of the object that contribute greater numbers of data points. Hence, the error function must be modified in such a manner that the contributions of different portions of the visible components remain unchanged when either the viewpoint or the image resolution is changed.

The generalized eigenvalue fit method used to derive the parameters of the surface involves two matrices that contain different terms of the form $\sum_{i=1}^{q} x^p y^q z^r$, where the summation is done over the set of available data points. Consider the contributions of four data point that are immediate neighbors of each other in the range image. If we introduce five equally spaced data points between these points, the contribution of these nine points will be different from that of the original four. Thus, if we change the resolution of the range image or the contributions of individual portions of the visible components, the matrices involved in the generalized eigenvalue fit change, resulting in a corresponding change in the surface parameters. However, if we replace the summation of the terms over the four points by the *integration* of the terms over the area enclosed by these points in 3D space, then introducing five points between them and integrating over the smaller area so enclosed will not result in a change in the contribution of the nine points when compared to that of the original four points. Consequently, replacing all the summations with integrations results in invariance to both changes in image resolution and changes in viewpoint.

## 8.1 Integration scheme

The integration of terms in the matrices used in the generalized eigenvalue fit is done under the assumption that four neighboring points in the range image define a plane. However, instead of explicitly determining the parameters of such a plane and solving the resulting integration problem for individual terms, *numerical integration* is used whereby points are interpolated between the four original points and integration is reduced to summation of the terms contributed by these points, with each term being multiplied by a suitable weight. The number of points to be interpolated and the weights associated with them depend on the specific integration formula being used. Since the highest order of the terms in the matrices $M_{\mathcal{D}}$ and $N_{\mathcal{D}}$ is 4, we need an integration formula where the leading term of the residual error depends on the $5^{th}$ or higher derivative of the function being integrated since then the residual error will be zero for all the terms and will be independent of the step size used in the integration. The integration rule being used here is *Bode's Rule*, where the leading term of the residual error depends on the $6^{th}$ derivative of the function being integrated.

### 8.2 Bode's rule

Let the values of a function $f(x)$ be tabulated at points $x_i, i = 1, \ldots, 5$ equally spaced by $h = x_{i+1} - x_i$, so that $f_1 = f(x_1), \ldots, f_5 = f(x_5)$. The Bode's rule approximating the integral of $f(x)$ is given by the Newton-Cotes-like formula

$$\int_{x_1}^{x_5} f(x)dx = \frac{2}{45}h(7f_1 + 32f_2 + 12f_3 + 32f_4 + 7f_5) - \frac{8}{945}h^7 f^{(6)}(\xi) \tag{54}$$

where the residual error $\frac{8}{945}h^7 f^{(6)}(\xi)$ is dependent on the $6^{th}$ derivative $f^{(6)}$ of the function $f(x)$ and will be zero when applied to the moment terms of the matrices $M_{\mathcal{D}}$ and $N_{\mathcal{D}}$, where the highest order involved is 4.

### 8.3 Application of Bode's rule

Bode's rule can be applied in this context as follows:

1. Form a $5 \times 5$ grid on four neighboring points in the range image such that these points coincide with the corners of the grid and the remaining points are interpolated by assuming that the four points form a plane. This can be done by using the interpolation rule

$$p(a,b) = (1-a)(1-b)p_1 + a(1-b)p_2 + (1-a)bp_3 + abp_4 \tag{55}$$

   where $p_i, i = 1, \ldots, 4$ are the four original points and $a$ and $b$ take five equally spaced values between 0 and 1. Thus, 21 interpolated points together with the four original points are involved in the integration.

2. These 25 points are weighted by the $5 \times 5$ matrix $ww^t$ where $w = \frac{2}{45}\begin{pmatrix} 7 & 32 & 12 & 32 & 7 \end{pmatrix}^t$.

3. The step size $h$ is given by $h = \frac{area}{16}$ where $area$ is the area of the polygon enclosed by the original four points.

4. The final weight assigned to the 25 points is $W = hww^t$.

5. The matrices $M_{\mathcal{D}}$ and $N_{\mathcal{D}}$ are modified as

$$M_{\mathcal{D}} = \frac{1}{q^*}\sum_{i=1}^{q^*} W_i cov_i \tag{56}$$

$$N_{\mathcal{D}} = \frac{1}{q^*}\sum_{i=1}^{q^*} W_i jcov_i \tag{57}$$

   where $cov_i$ and $jcov_i$ are given by Equation 45 and Equation 51, and $W_i$ is the final weight assigned to point $p_i$. Note that the total number of data points has been increased to $q^*$ to include all the interpolated points.

### 8.4 Caveat

1. The replacement of summed terms by integrated ones in the matrices $M_{\mathcal{D}}$ and $N_{\mathcal{D}}$ is done under the assumption that neighboring data points define a plane. While this would approximately hold when the data points actually define a plane or a low-order surface, it has serious consequences when the neighboring data points in the range image actually belong to different components of the object. Not only would we be integrating over non-existent portions of the object, we would also be considering very different portions for different viewpoints. Thus, it is necessary to identify those regions in the range image where there is a transition from one component of an object to another. Note that here we need to identify local discontinuities and not an entire homogeneous region which is a more difficult task. Once such discontinuities have been identified, we can disregard the contributions from these portions of the image. Since we need to fit planes in local neighborhoods in order to isolate the object from the ground, we can use the same information in order to identify portions of the range image where the plane fit causes large fitting errors. We can specify a loose threshold so as to incorporate noisy data and low-order surfaces and mark out those points where the fitting errors are large. These points will generally correspond to regions in the range image where there are transitions from one component of the object to another.

# 9

# Canonical Frame Mapping

This section describes how the quadratic surface extracted from the extended neighborhood of a data point can be used to define a 3D canonical frame of reference so that the coordinates of the data point can be mapped to the canonical frame to form invariant coordinates.

## 9.1 Standard scheme

This scheme involves determining a transformation that maps the quadratic surface to one whose origin and principal axes coincide with the origin and principal axes of the canonical frame. This transformation can then be used to map data points to this canonical frame.
For a quadratic surface defined by

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0 \tag{58}$$

define the following

$$e = \begin{pmatrix} a & h & g \\ h & b & f \\ g & f & c \end{pmatrix} \quad \text{and} \quad E = \begin{pmatrix} a & h & g & p \\ h & b & f & q \\ g & f & c & r \\ p & q & r & d \end{pmatrix} \quad \text{or} \quad E = \begin{pmatrix} e & s \\ s^t & d \end{pmatrix} \tag{59}$$

where $s = \begin{pmatrix} p & q & r \end{pmatrix}^t$.
If $X = \begin{pmatrix} x & y & z & 1 \end{pmatrix}^t$, then the quadratic surface can be represented as $X^tEX = 0$. We seek a rigid body transformation with parameters $(R, t)$ such that the surface is mapped to the standard form $\hat{X}^t\hat{E}\hat{X} = 0$ where

$$\hat{E} = \begin{pmatrix} \hat{e} & 0 \\ 0^t & \hat{d} \end{pmatrix} \quad \text{and} \quad \hat{e} = \begin{pmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & \lambda_z \end{pmatrix} \tag{60}$$

where $\lambda_x$, $\lambda_y$, and $\lambda_z$ have to be determined.
If $X = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \hat{X}$ is a particular Euclidean transformation, then the transformed surface is given by

$$\hat{X} \begin{pmatrix} R^t & 0 \\ t^t & 1 \end{pmatrix} E \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \hat{X} = 0 \tag{61}$$

Thus

$$\begin{aligned} \hat{E} &= \begin{pmatrix} R^t & 0 \\ t^t & 1 \end{pmatrix} \begin{pmatrix} e & s \\ s^t & d \end{pmatrix} \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} && (62) \\ &= \begin{pmatrix} R^t & 0 \\ t^t & 1 \end{pmatrix} \begin{pmatrix} eR & et + s \\ s^tR & s^tt + d \end{pmatrix} && (63) \\ &= \begin{pmatrix} R^teR & R^t(et + s) \\ (R^t(et + s))^t & t^t(et + s) + (s^tt + d) \end{pmatrix} && (64) \end{aligned}$$

which leads to the conditions

$$\hat{e} = R^teR, \quad et + s = 0 \quad \text{and} \quad \hat{d} = s^tt + d \quad \text{since} \quad et + s = 0 \tag{65}$$

If the SVD decomposition of the symmetric matrix $e$ is given by

$$e = U \Lambda U^t \tag{66}$$

then since $\hat{e} = R^t e R$ or $e = R \hat{e} R^t$, we observe that

$$R \quad = \quad U \quad \text{and} \quad \hat{e} \quad = \quad \Lambda \tag{67}$$

where $\Lambda$, being a diagonal matrix, is of a form suitable for $\hat{e}$. Similarly, the translation $t$ is given by

$$t \quad = \quad e^{-1} s \tag{68}$$

and the required transformation becomes

$$\hat{X} \quad = \quad \begin{pmatrix} R^t & -R^t t \\ 0 & 1 \end{pmatrix} X \tag{69}$$

## 9.2 Caveat

With the exceptions of neighborhoods that essentially define a plane, most extended neighborhoods in the range image define quadratic surfaces one of whose principal axes is close to infinitely long. This implies that one of the singular values of $e$ is often close to zero, so the calculation of the inverse $e^{-1}$ is prone to error. Thus, in such cases the calculation of the translation parameter $t$ could yield very erroneous results. If instead we calculate the pseudo-inverse, we could constrain the origin of the transformed quadratic surface to lie on a line passing through the origin in the canonical frame, but the position along this line would still remain unresolved.

## 9.3 Modified scheme

This seeks to derive a usable canonical frame even in cases where one of the principal axes of the surface may be close to infinitely long. This is the situation which is encountered most often in a range image, especially when the extent of the neighborhood is small.
The canonical frame is constructed as follows:

1. The centroid of the points in the specified neighborhood, rather than the origin of the quadratic surface, is assumed to map to the origin of the canonical frame. The centroid is a weighted mean of the coordinates of the points, the weights being those associated by using the Bode's rule on the terms $\frac{1}{q}\sum_{i=1}^{q} x_i$, $\frac{1}{q}\sum_{i=1}^{q} y_i$, and $\frac{1}{q}\sum_{i=1}^{q} z_i$ and thus the position of the centroid is not dependent on the resolution of the data points.

2. The three eigenvectors of $e$ obtained as a result of its SVD are assumed to coincide with the coordinate axes of the canonical frame. If one of the singular values is close to zero, its eigenvector can still be used since it will be the cross-product of the two eigenvectors corresponding to the eigenvalues with the higher magnitudes.

3. The three eigenvectors can be mapped to the coordinate axes of the canonical frame in any order. In order to enforce a consistency in this mapping, the order is determined by the magnitude of the corresponding eigenvalues. For instance, the eigenvector corresponding to the largest eigenvalue is

mapped to the $x$-axis of the canonical frame, that corresponding to the next smaller eigenvalue is mapped to the $y$-axis, and so on.

4. Before this mapping is done, the eigenvectors are modified so that the specified order defines a right-handed system. This can be done since both $v$ and $-v$ can be eigenvectors corresponding to the same eigenvalue.

5. Although the modified scheme can be used to derive a canonical frame in most cases, there are certain ambiguities in the mapping resulting from the heuristics used in specifying the mapping order. The heuristic used to order the mapping of the eigenvectors to the coordinate axes of the canonical frame is based on the magnitudes of the corresponding eigenvalues. However, there remains an ambiguity as regards the directions of the coordinate axes since both $v$ and $-v$ can be eigenvectors for the same eigenvalue. Although enforcing the right-handedness constraint discards some of the possibilities, some combinations are still not resolved.

6. It is possible to resolve these ambiguities by imposing constraints on some global characteristics of the points enclosed in the neighborhood. The constraint being used currently ensures that a greater number of data points lies in the positive half-spaces of the two eigenvectors with the largest two eigenvalues. In other words, if $\{p_i\}_{i=1}^{q}$ are the data points, $cen$ their weighted centroid, $v_x$ and $v_y$ the eigenvectors corresponding to the two largest eigenvalues of the matrix $e$, and $n(S)$ the cardinality of the set $S$, then the following condition must hold:

$$\begin{array}{rcl} n(sg((p_i - cen) \cdot v) \geq 0) & \geq & n(sg((p_i - cen) \cdot v) = -1) \\ & & \text{for } v = v_x \text{ and } v_y,\ i = 1, \ldots, q \end{array} \tag{70}$$

   where $sg(x)$ is the *signum* function and $\cdot$ represents the dot product. If this condition does not hold for a eigenvector $v$, set $v = -v$. The direction of the third eigenvector is chosen as the one which fulfills the constraint that the three eigenvectors should define a right-handed system.

# 10

# Matching

In this section, we describe how the dataset of model points is developed and how the search for a correspondence with a test range image is carried out. Before describing either of these, it is worthwhile to elucidate how a set of points are extracted from a given range image and mapped to the canonical frame, since the identical procedure is used both for model and test range images.

## 10.1 Determining plane discontinuities

This step attempts to localize regions in a range image where there is a transition from one component of the object to the other. This is necessary while calculating the matrices $M_{\mathcal{D}}$ and $N_{\mathcal{D}}$ (see Section 8.3) in order to prevent integration of terms over non-existent portions of the object. Consequently, such discontinuities have to be located before any invariant characteristic can be determined. The procedure used is described below:

1. Fit a plane to the $3 \times 3$ neighborhood of each point of the range image and estimate the errors involved in each such fit.

2. Mark all the points where this fitting error exceeds a specified threshold. This threshold is kept quite loose in order to include noisy data and low-order surfaces. Thus, if $I(x, y)$ is a range image where each pixel $(x, y)$ contains the 3D coordinates of a data point, then we create a 2D mask $M(x, y)$ such that $M(x, y) = 1$ if the error exceeds the threshold and is zero everywhere else.

3. If the particular range image is a test image, then this plane fitting procedure can be used to isolate the object from the background. If the background is assumed to be approximately planar and occupies a larger portion of the range image than any other single plane, then we can segment all the data points into approximate planes and purge out the points corresponding to the largest such plane if the area covered by it shows a marked dominance over the areas covered by the rest of the planes. Note that these planes are being used solely to remove the background and not for the actual recognition process. Consequently, any errors in the estimation of their parameters are not of great significance.

4. Mark all the points on the range image such that they and their immediate neighbors do not lie on the plane discontinuities thus found. Only these points and their neighbors are used in the calculation of invariant characteristics.

## 10.2 Mapping to the canonical frame

Once the regions containing component transitions have been identified, we can extract invariant characteristics from the range image. Instead of determining characteristics for each data point available, we employ only a subset. The entire mapping process can be summarized as follows:

1. Extract a set of data points by sampling the range image with a uniform grid. The resolution of the grid is unimportant as long as it is reasonably dense. If any of the grid points happens to fall on a region mark it as a transition between object components, calculate the nearest point $(x, y)$ with $M(x, y) = 0$ and take this point as the grid point. The question of grid resolution is discussed in greater detail in Section 10.4.

2. The procedure described below is used to extract a canonical frame for each such data point at a particular value of the scale parameter which in this context corresponds to a particular value of the neighborhood radius. The particular subset of radius values used from the pre-defined set of radius values differs if the range image concerned is a model or a test image:

   - In the case of a model range image, all the radius values in a contiguous subset are used. The extrema of this subset are obtained as follows:
     - The minimum is determined by the radius at which the quadratic surface fitted to the neighborhood centered on at least one data point corresponds to coincident planes. In other words, we should be able to extract usable canonical frames from all the sampled data points at all radii.
     - The maximum is one where the entire object is included in the radius around each of the sampled data points and increasing the radius value would be meaningless since new components will never be included in the expanding neighborhood.
   - In the case of a test range image, we start with the largest possible radius value and proceed towards the smallest. If for two consecutive radii, we obtain a match for the same model with reasonable reliability, we can terminate the matching process.

3. The procedure for determining the invariant characteristics of a data point at a particular radius value is described below:

   (a) Determine all the marked points within the sphere centered at the data point with the specified radius. These points are such that they and their neighbors do not lie on plane discontinuities.

   (b) Assuming that each such point and its immediate neighbors define a plane, determine the area enclosed by the polygon formed by these points and the coordinates of the additional 21 interpolated points required for the application of Bode's rule (see Section 8.3).

   (c) Calculate the matrices $M_{\mathcal{D}}$ and $N_{\mathcal{D}}$ as described in Section 8.3.

   (d) Determine the parameters of the quadratic surface $\alpha$ as the eigenvector corresponding to the smallest eigenvalue of the linear system

   $$M_{\mathcal{D}} x = \lambda N_{\mathcal{D}} x$$

   (e) Determine the coordinates of the weighted centroid and the eigenvectors of the matrix $e$ formed from $\alpha$ as described in Section 9.3.

   (f) The centroid and the eigenvectors serve to determine the parameters of the canonical frame. Determine the transformation from the coordinate space of the data points in the range image to the canonical frame, as described in Section 9.3.

(g) Remove the ambiguities in the mapping by enforcing the following constraints on the directions of the eigenvectors:

- The directions of the eigenvectors corresponding to the two largest eigenvalues should be such that a larger fraction of the points in the neighborhood lie in the positive half-space of each eigenvector (see Section 9.3).
- The direction of the third eigenvector is determined as one that makes the three eigenvectors a right-handed system.

(h) Using this transformation, map the data point to the canonical frame. Its coordinates in the canonical frame constitute the invariant coordinates.

## 10.3 Model dataset

1. Each model is represented in terms of the invariant characteristics generated from a number of range images corresponding to different viewpoints. Images from two different viewpoints are required only when considerably different portions of the model are visible in the two views.

2. The data points in each range image are sampled using a grid of fixed resolution. For each of these data points, invariant characteristics at each of a specified set of scale parameter values are determined as described in Section 10.2.

3. If we collect the invariant characteristics of each grid point at a particular scale, we obtain a collection of 3D points that are representative of the entire model at a particular scale. This set of points can be characterized by identifiers for the model, the scale and the viewpoint from which they are extracted, and can be considered as a single entity for purposes of matching.

4. Since during matching, we need to determine the model data points lying in the vicinity of a data point extracted from a test range image, a hierarchical spatial organization of the model data points such as an oct-tree or even an R-tree would prove useful. However, the question of organizing the model data points is addressed in Section 14 when a framework for handling articulation is incorporated into the matching process. For now, it is sufficient to assume that there exists some mechanism to extract the points lying in the neighborhood of a test data point in the canonical frame, possibly by an exhaustive search of all the points.

## 10.4 Search

Before describing the search procedure, it is necessary to define a measure of goodness for a partial correspondence that can serve to discriminate between different matches. Ideally, such a measure would be independent of the canonical frame mapping since it can serve as a totally distinct test and not be hampered by the ambiguities in the mapping process.

### Disparity measure

This measure captures the spatial relationship of a set of points and can be used to compare the actual coordinates of a set of data points from a test range image and the coordinates of the corresponding model data points.

Let $k$ be the cardinality of a set of points $P$. Let $D_{k \times k}$ be the relative distance matrix, with $D(i,j)$ being defined as the Euclidean distance between $P_i$ and $P_j$. The *disparity measure* $d$ is the degree of correlation between the relative distance matrices derived from the image data points and their corresponding model data points and is defined as

$$d = \sqrt{\frac{\sum_{i,j}(D_{image}(i,j) - D_{model}(i,j))^2}{k^2}} \tag{71}$$

The disparity measure has the following properties:

1. Since it is based on the actual coordinates of the data points as obtained from the range images and not on the invariant coordinates in the canonical frame, it is independent of the mapping used to transform the data points to the canonical frame.

2. Since it is based on the distances between points rather than on the actual coordinates of the points, it is independent of the coordinate frames in which the coordinates of the model and image data points are expressed.

## Partial correspondence

In the model dataset, each data point in the canonical frame is indexed by a tuple (*model*, *view*, *radius*) where *model* is an identifier for the corresponding model, *view* is an identifier for the viewpoint from which the range image was created, and *radius* is the radius of the neighborhood sphere from which the canonical frame was defined.

Thus, for a particular sample of data points from a test range image, partial correspondence sets are created for each distinct (*model*, *view*) combination present in the model dataset. The size of each set is equal to the cardinality of the sample. Each slot in the set is used to store an identifier for the corresponding model point, for each sample data point. All the slots are initially empty and get occupied as the search procedure progresses. The measure of goodness for a partial correspondence set is the disparity measure weighted by the number of non-empty slots in the set.

## Search procedure

1. Sample the test range image using a uniform grid and obtain a set of data points whose invariant characteristics will be considered.

2. Create empty partial correspondence sets for each (*model*, *view*) combination with cardinalities equal to the number of image data points extracted.

3. Consider a range of radius values, starting from the largest possible. The minimum value of the radius is one at which any one of the neighborhoods centered on the sampled data points does not define a usable canonical frame.

4. For each radius, do:

   (a) For each data point, do:

   - *Mapping*: Transform the data point to the canonical frame using the procedure described in Section 9.3.

- *Finding neighbors*: If the coordinates of the data point in the canonical frame are $(x, y, z)$, consider all the model data points within a specified distance from $(x, y, z)$ whose *radius* values match the current radius.
- *Initial matches*: From the set of retained points, find the closest points with distinct (*model*, *view*, *radius*) values.
- *Modified matches*: Using the canonical frame defined by each such closest point, transform all the model range data points in the neighborhood of that point to the canonical frame and determine if there is a point that is closer to the data point derived from the test range image. The neighborhood in this context includes all the data points between that point and the neighboring points on the grid used to sample the model data points.

(b) Collect the closest model points so obtained for each of the sampled image data points, and form temporary partial correspondence sets for each distinct $(model, view)$ present in the entire collection of candidate matches. Update the original partial correspondence sets by replacing those sets whose disparity measures are inferior to the corresponding ones in these temporary sets.

(c) If the partial correspondence set belonging to some $(model, view)$ has reasonably good disparity measures for two consecutive radii, record the $(model, view)$ in the final set of hypothetical matches. Terminate the search as soon as a specified number of such matches have been obtained.

5. Verify each of these hypothetical matches by directly calculating the transformation between the model and test range image from the partial correspondences, projecting the model range image onto the test image and performing a per-pixel comparison. The model that yields the best correlation is deemed to correspond to the object in the test image.

## Grid resolutions

It is important to note that the procedure described above does not depend on the actual resolution of the grid used to sample the model and test range image. This is achieved through the following:

1. A sufficiently dense grid is used to sample the model range image in order to ensure that the canonical frame defined by a model data point at a particular radius will be approximately the same as that defined by the points between this point and the neighboring grid points. This implies that all these points can be mapped to the canonical frame defined by the grid point with sufficient accuracy.

2. Having found the model point closest to a mapped data point from the test range image in the canonical frame, all the points in the neighborhood of the model data point are mapped to the canonical frame to investigate if a closer match can be found.

3. The grid with which the test range image is sampled can be much coarser than that used to sample the model range image. This is because each individual data point from the test range image contributes separately to the final match, and a large number of data points are not needed in order to find a match of sufficiently high reliability.

# 11

# Results

In this section, we report the results of experiments conducted with a prototype implementation of the recognition system described in the previous sections. The experiments were conducted with synthetic range data using real-life models. In order to verify that view-invariance is indeed achieved, we attempt to recognize objects in range images where the viewpoints involved are very different from the viewpoints from which the corresponding model range images were obtained.

The model dataset consists of five range images, one each for five models, most of them military vehicles. The model range images are shown in Figure 1, and their specifications are listed in Table 1. Invariant characteristics are generated from points sampled from each of these range images using a reasonably dense uniform grid. For each point, these characteristics are obtained for three different neighborhood radii. Even though a single radius could be sufficient for matching, more than one value is necessary in order to achieve reasonable match reliability. The largest radius is chosen to be one that encompasses the entire object.

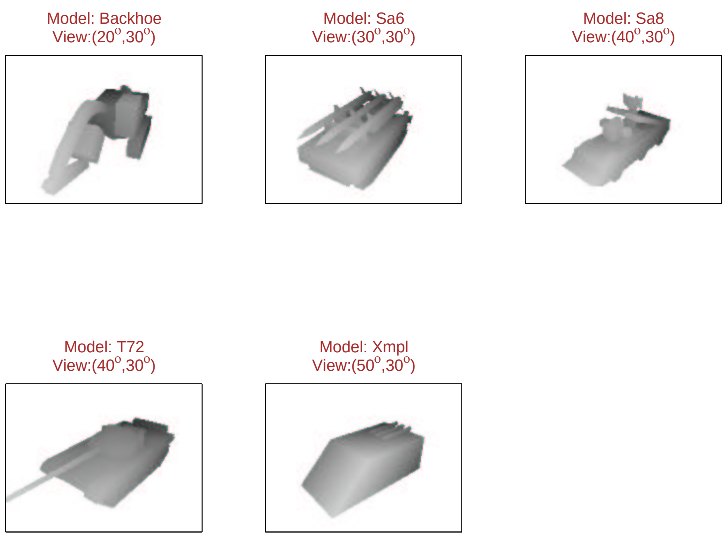


Figure 1: Models

| Model Name | Image Size (width,height) | Viewpoint (azimuth,elevation) |
|---|---|---|
| BACKHOE | (400,300) | $(20°, 30°)$ |
| SA6 | (400,300) | $(30°, 30°)$ |
| SA8 | (400,300) | $(40°, 30°)$ |
| T72 | (400,300) | $(40°, 30°)$ |
| XMPL | (400,300) | $(50°, 30°)$ |

Table 1: Model specifications

| Model Name | Model Viewpoint (azimuth,elevation) | View | Image Size (width,height) | Viewpoint (azimuth,elevation) |
|---|---|---|---|---|
| BACKHOE | $(20°, 30°)$ | View 1 | (200,150) | $(10°, 30°)$ |
| | | View 2 | (200,150) | $(30°, 30°)$ |
| SA6 | $(30°, 30°)$ | View 1 | (200,150) | $(20°, 30°)$ |
| | | View 2 | (200,150) | $(45°, 30°)$ |
| SA8 | $(40°, 30°)$ | View 1 | (200,150) | $(20°, 30°)$ |
| | | View 2 | (200,150) | $(75°, 30°)$ |
| T72 | $(40°, 30°)$ | View 1 | (200,150) | $(20°, 30°)$ |
| | | View 2 | (200,150) | $(60°, 30°)$ |
| XMPL | $(50°, 30°)$ | View 1 | (200,150) | $(30°, 30°)$ |
| | | View 2 | (200,150) | $(80°, 30°)$ |

Table 2: Specifications for the test range images

The test images include two views of each object in the model dataset. Note that we are attempting to match the object in both these views to the same model range data. The specifications for the views are listed in Table 2. Observe that there is a wide disparity between the viewpoints of the test range images and the viewpoint of the corresponding model range data. Moreover, in the test range images, the objects have been placed on a planar background, so that the range data for the test images also include data for the background.

Recognition is carried out by sampling the test range images using a relatively coarse grid and extracting invariant characteristics for each grid point for each of the neighborhood radii in the pre-defined set. For an individual invariant point in these sets, we determine the closest invariant point for each model. The results of the recognition process are shown in figures of three different types:

1. Figures with correct models (Figures 2, 5, 8, 11, 14, 17, 20, 23, 26 and 29): These figures illustrate the matching between the test range images and the model range images for the correct model at different neighborhood radii. The images in the first column show the grid points chosen and representations of the extent of each neighborhood from which the invariant characteristics are derived. The images in the first row correspond to the case when the neighborhood encompasses the entire object. The images in the second column show the closest model invariant point found for each grid point, with points having the same color being the matches. Observe that most of the model invariant points found are reasonably close to the correct corresponding model point. The images in the third column show the projections of the model onto the test range images which, in most cases, are relatively close to the objects in the test images.

2. Figures with incorrect models (Figures 3, 6, 9, 12, 15, 18, 21, 24, 27 and 30): These figures illustrate the fact that incorrect models do not match the objects in the test range images. The images in the first column show the grid points chosen in the test range images. The grid points chosen in all the images are identical. The neighborhood radius considered in each case is one that encompasses the entire object. The images in the second column show the closest model invariant point found for each grid point, with points having the same color being the matches. Observe that the matches are either randomly spread all over the model or concentrated in a small portion. The images in the third column show the results of projecting the model onto the test range image and demonstrate the

disparity between the projections and the actual objects in the test images.

3. Figures showing the relative disparities (Figures 4, 7, 10, 13, 16, 19, 22, 25, 28 and 31): The disparity measure, given by Equation 71, estimates the degree of correlation between the grid points on the test range image and their matching model points, and should be the smallest when the matching is done with the correct model. The relative disparity of a model is the ratio of the value of the disparity measure when the correct model is considered, to the value of the disparity measure when the particular model is considered. Matching can be said to be correct when the relative disparity for the correct model (equal to 1) is greater than the relative disparities for all the other models. The figures demonstrate that correct recognition occurs for all the test range images considered.

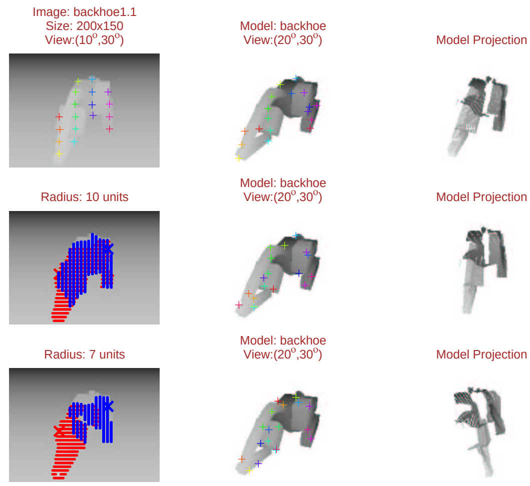


Figure 2: Model BACKHOE, View 1: Comparison to the correct model

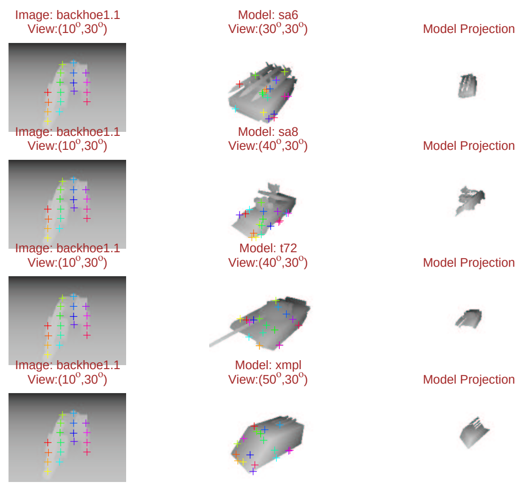


Figure 3: Model BACKHOE, View 1: Comparison to incorrect models

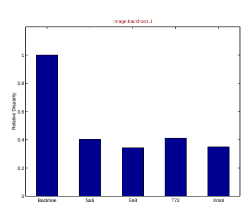

Figure 4: Model BACKHOE, View 1: Relative disparities w.r.t. all models

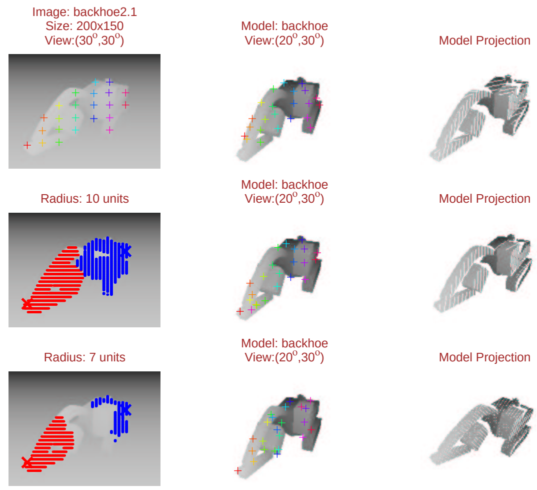


Figure 5: Model BACKHOE, View 2: Comparison to the correct model

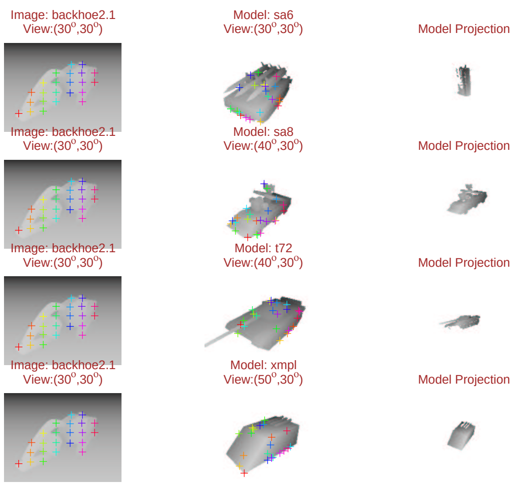


Figure 6: Model BACKHOE, View 2: Comparison to incorrect models

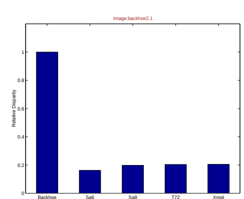


Figure 7: Model BACKHOE, View 2: Relative disparities w.r.t. all models

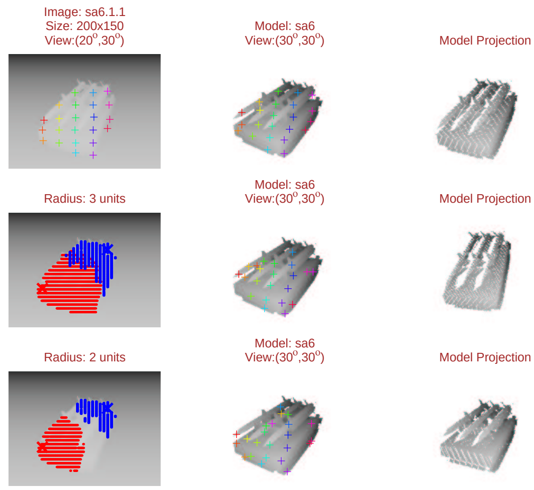


Figure 8: Model SA6, View 1: Comparison to the correct model

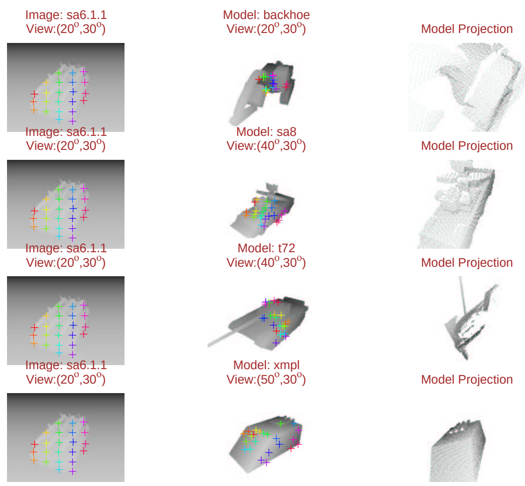


Figure 9: Model SA6, View 1: Comparison to incorrect models

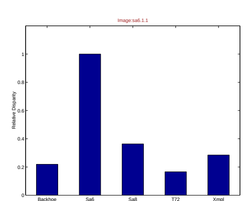


Figure 10: Model SA6, View 1: Relative disparities w.r.t. all models

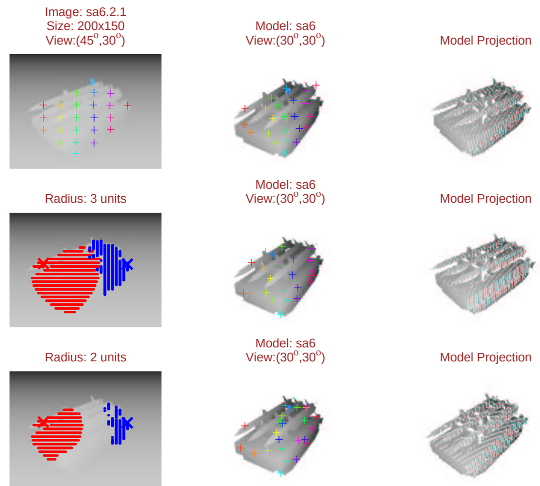


Figure 11: Model SA6, View 2: Comparison to the correct model

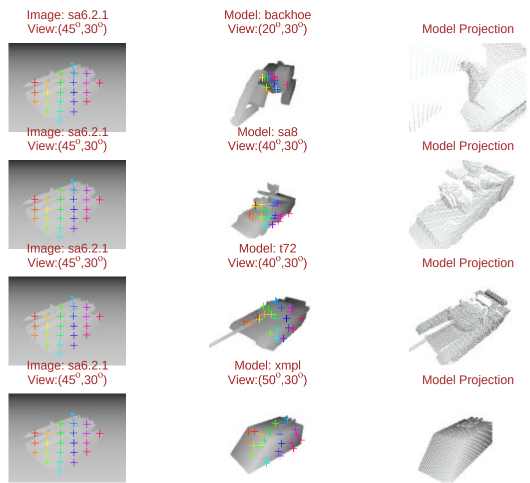


Figure 12: Model SA6, View 2: Comparison to incorrect models

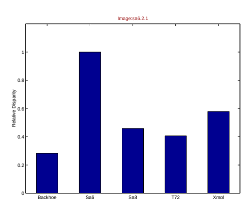


Figure 13: Model SA6, View 2: Relative disparities w.r.t. all models

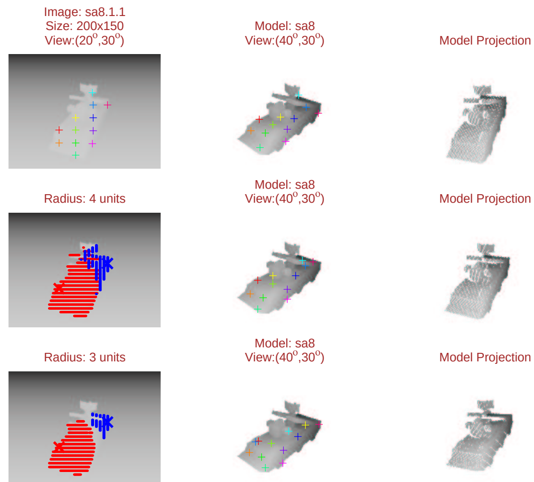


Figure 14: Model SA8, View 1: Comparison to the correct model

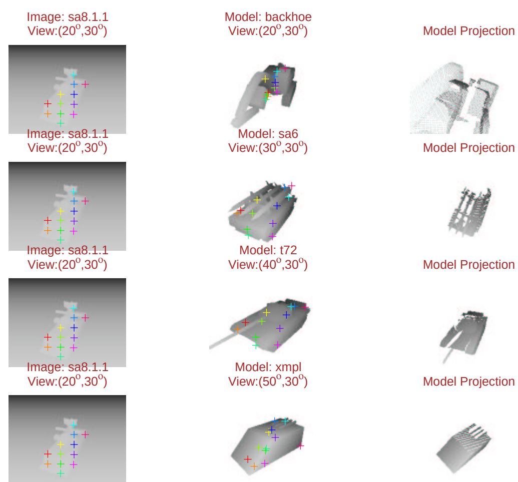


Figure 15: Model SA8, View 1: Comparison to incorrect models

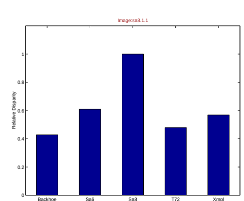


Figure 16: Model SA8, View 1: Relative disparities w.r.t. all models

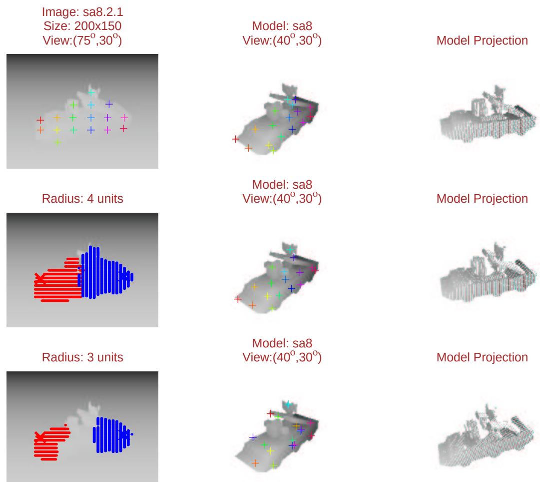


Figure 17: Model SA8, View 2: Comparison to the correct model

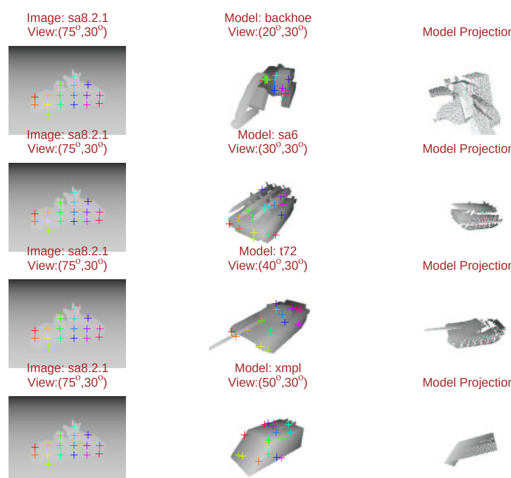


Figure 18: Model SA8, View 2: Comparison to incorrect models

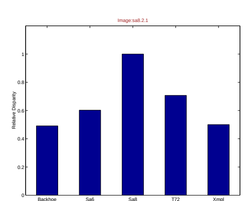


Figure 19: Model SA8, View 2: Relative disparities w.r.t. all models

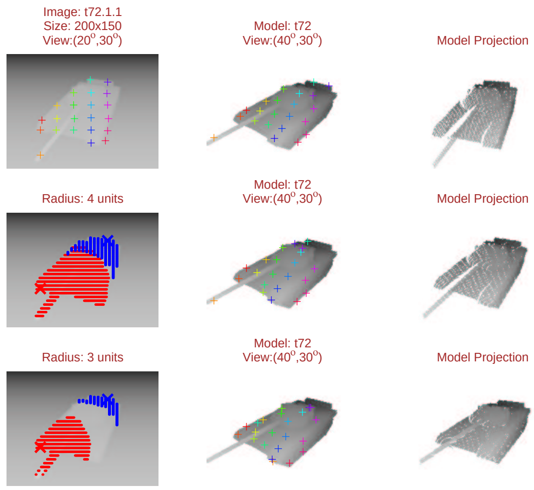


Figure 20: Model T72, View 1: Comparison to the correct model

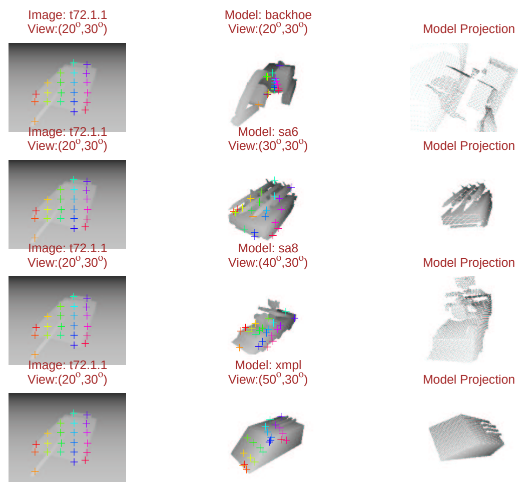


Figure 21: Model T72, View 1: Comparison to incorrect models

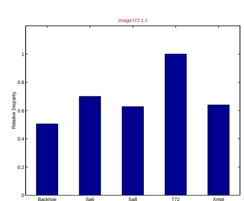

Figure 22: Model T72, View 1: Relative disparities w.r.t. all models

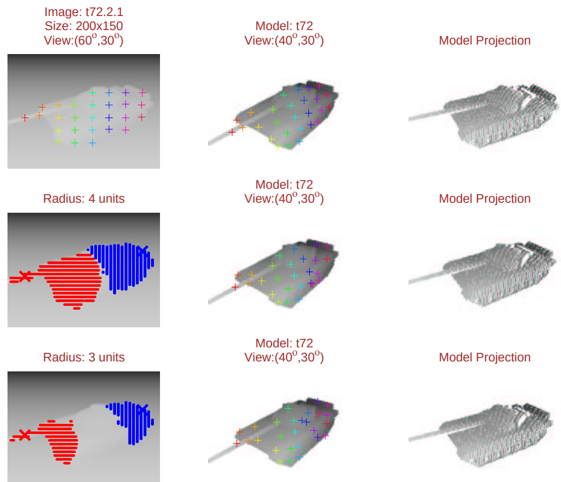


Figure 23: Model T72, View 2: Comparison to the correct model

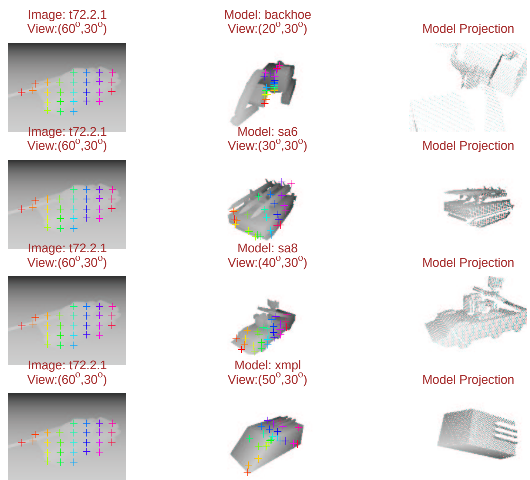


Figure 24: Model T72, View 2: Comparison to incorrect models

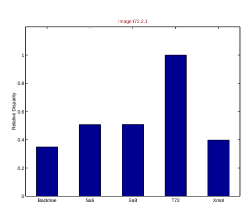


Figure 25: Model T72, View 2: Relative disparities w.r.t. all models

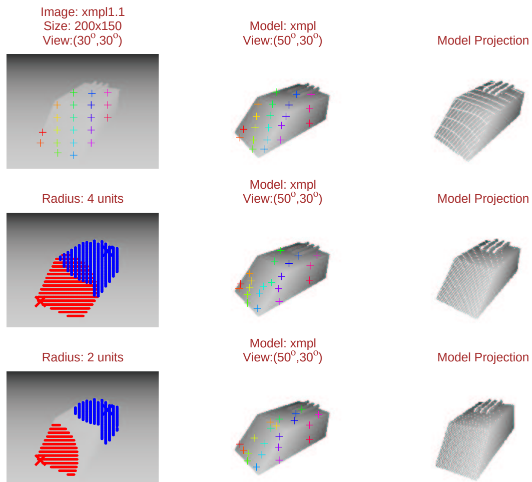


Figure 26: Model XMPL, View 1: Comparison to the correct model

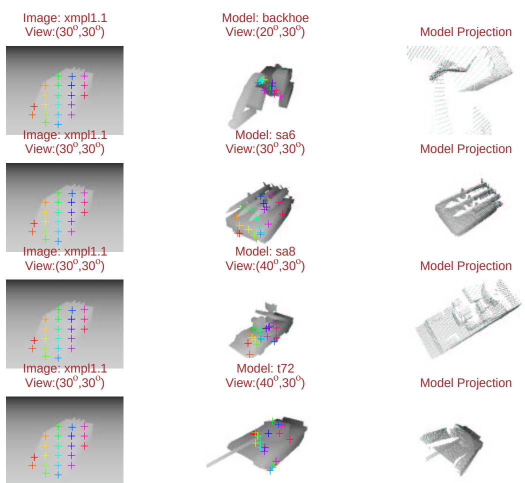


Figure 27: Model XMPL, View 1: Comparison to incorrect models

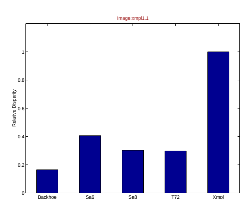

Figure 28: Model XMPL, View 1: Relative disparities w.r.t. all models

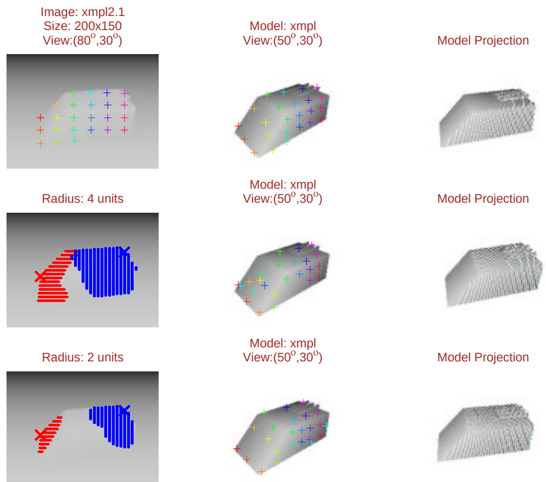


Figure 29: Model XMPL, View 2: Comparison to the correct model

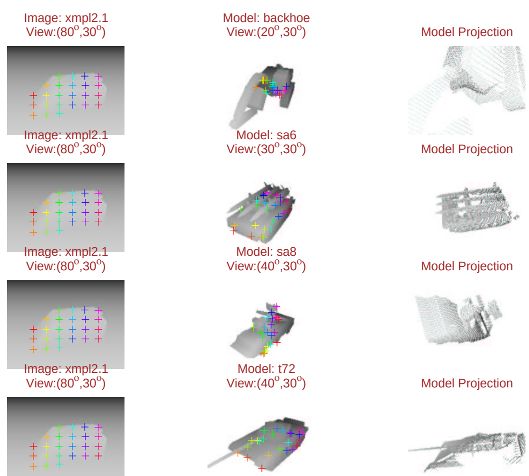


Figure 30: Model XMPL, View 2: Comparison to incorrect models

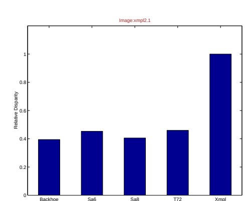

Figure 31: Model XMPL, View 2: Relative disparities w.r.t. all models

# 12

# Recognition of Articulated Objects: Introduction

In the following sections, we extend the framework to incorporate articulation in the objects of interest. Articulated objects are those that have components attached via joints and can move with respect to each other. This deformation makes the task of recognition substantially more difficult since in general the articulation of the object in the range image will be different from that of the model in the dataset and it will be too prohibitive to consider each possible articulated model as a distinct one. However, a significant characteristic of articulated objects is that they only have limited numbers of degrees of freedom (DOF), and hence it is possible to develop a mechanism whereby these DOFs are included in the information associated with the model and a particular configuration of these DOFs in the range image can be identified.

## 12.1 Background

Different approaches have been proposed for the recognition of articulated objects. Many of these are based on the ability to extract geometric representations of portions of the object from individual features detected in the image. The naive method is to localize each component separately before determining the inner joint states. This approach neither exploits the constraints imposed by different joint types nor can it deal with self-occlusion. The more formal solution, the extension of the aspect graph concept by object configurations [Reference 16], leads to an explosion of the number of possible aspects even in simple cases. Global parametric methods [Reference 12] simultaneously estimate the poses of all object components, which can result in a very large search space for correspondence between image and model features.
In [Reference 7], a hierarchical representation of the object as a composition of rigid components which are explicitly connected by specific kinematic constraints is used and the recognition task follows a tree-like structure by first estimating the pose of the static component (root) and afterwards determining the relative poses of the remaining components recursively. However, this generic mechanism of identifying components by aggregating features satisfying a set of constraints is dependent on a feature extraction process and a subsequent grouping algorithm, both of which can be much prone to errors.
A previous work concentrating on recognizing articulated objects in range images using invariance [Reference 25] was based on surface segmentation using normals. Planar surface patches were identified in the range images, and patches with similar normals were grouped together. Given the normals, the relative angles between them can be determined, which though invariant to similarity transforms, is not invariant to articulations. Consequently, all possible angles corresponding to all possible configurations of the DOFs of a model were stored in a hash table and recognition was achieved by determining the angles from the test range image and performing a lookup in the hash table to determine the model, its pose and articulation. Although the surface normals can be made to behave like global similarity relative invariants by averaging over the entire surface patch and are thus less sensitive to noisy data points, the angles between normals are not articulation-invariant and hence all the angles from all possible configurations have to be stored, which is feasible only for a limited number of degrees of freedom.

# 13

# Recognition of Articulated Objects: Overview

The salient features of the procedure described in the previous sections for recognizing objects in single range images using scale-space invariants are:

1. The range image was sampled with a uniform grid to obtain a set of individual data points.
2. Invariant coordinates were obtained for each data point by mapping it to the canonical frame at each discrete value of the scale parameter.
3. The collection of invariant points in the canonical frame for a particular scale corresponding to all the sampled data points provides a complete description of the model at the specified scale. These points in general define a surface in the 3D canonical frame. If the scale parameter is large enough that the neighborhoods of each of the sampled data points include the entire object, this surface will closely resemble the model provided the grid resolution used is reasonably dense.
4. The matching algorithm derives invariant points in the canonical frame from sampled data points of the test range image for different scale levels and then determines the surfaces on which these invariant points lie. However, since the points will lie exactly on the surfaces only occasionally, the algorithm is designed to find points on the surfaces that are closest to the test invariant points.

The objective of this work is to extend this mechanism so as to make the task of recognizing articulated objects feasible. The constraints imposed on this task are:

1. The framework described in the previous sections should remain applicable for objects that have no articulated components.
2. The recognition task should not depend on the extraction of discrete features from the range image or on the grouping of these features in an effort to localize individual components.
3. Even if all possible configurations of the articulated components are examined while developing the representation for a model, the final representation should not grow prohibitively as the number of possible degrees of freedoms is increased.
4. During matching, when it is necessary to determine the intersections of the subspaces generated from the invariant data points extracted from the test range, with the invariant subspace representation of a model in the canonical frame, it should be possible to do so by examining the representation only in the vicinity of the points of interest rather than an extended portion of it.

## 13.1 Articulation invariants

We face a significant problem when we attempt to derive invariant characteristics for models with articulated components. Invariant characteristics generally apply to particular transformations; for instance, if an object is viewed from different viewpoints, we attempt to derive viewpoint invariants assuming differing types of model invariants and image projection modes. Specific forms of viewpoint invariants involve well-defined groups of transformations, for example Euclidean transformations in the case of range images, and can be applied to any object we want to recognize, i.e. the transformation group is independent of the model. Thus, these viewpoint invariants can be applied generically to all models. However, in the case of articulated objects, this is no longer the case. Each model has its own set of articulation degrees of freedom which translates into its own specific transformation group. It is difficult to determine the degrees of freedom of an object from its range image, and hence its transformation group remains unknown. Therefore, while it is theoretically possible to derive articulation-invariant functions for each individual model, it is difficult to determine generic invariants that can be applied to all models. Since we cannot employ real articulation invariants, we attempt to incorporate articulation in the canonical frame method developed for the recognition of unarticulated objects from their range images.

## 13.2 Hyper-surface representation

Recall that in the method described in the previous part, data points on a grid were mapped to invariant points in the canonical frame for a particular value of the scale parameter, and if we consider the invariant coordinates of all the data points at that particular scale value, they define a 3D surface in the canonical frame. If the object from which these data points were derived had articulated components, then this surface would correspond to a particular configuration of the articulated components – in other words, to a particular value of the articulation parameters. Changing the articulation parameters results in a different surface in the canonical frame. If we increase the dimensionality of the canonical frame by adding a dimension for each distinct articulation parameter, then the surfaces generated by all possible values of the articulation parameters can be combined into a single hyper-surface.

Denoting the invariant coordinates of a data point at a particular value of the scale parameter by $\mathbf{x}$, we note that they are functions of the articulation parameters $\mathbf{u}$, i.e. $\mathbf{x}(\mathbf{u})$. This relationship can be expressed as the implicit hyper-surface $f(\mathbf{x}, \mathbf{u}) = 0$. To construct this hyper-surface, we vary the articulation parameters, and for each value $\mathbf{u}$ we determine all the corresponding coordinates $\mathbf{x}$.

The principal motivation behind using this representation is the realization that there exists a great deal of correlation between the surfaces generated by the sampled data points at two values of the articulation parameters that are close to each other. In other words, the hyper-surface will be relatively smooth. Consequently, it should be possible to come up with a compact representation of the surface that does not grow prohibitively as the number of articulation parameters increases. The representation implemented in this work uses multi-dimensional wavelets.

### Representation

It is necessary to derive a discrete representation of the hyper-surface $f(\mathbf{x}, \mathbf{u}) = 0$ in order to use discrete multi-dimensional wavelets to compress it. To do so, we discretize each dimension of the surface, and mark the voxels lying on the surface $f$ by 1 and leave the other voxels as 0. This is done since the surface, in general, can only be represented implicitly and it may not be possible to come up with an explicit representation of the surface for all possible models.

## Indexing

Indexing amounts to inverting the function $\mathbf{x}(\mathbf{u})$, i.e. we are given the invariant coordinates $\bar{\mathbf{x}}$ of a data point extracted from a test range image and we want to find the articulation parameters that might correspond to $\mathbf{u}(\bar{\mathbf{x}})$. Using the hyper-surface representation we obtain a form of indexing in which, given the invariant coordinates $\bar{\mathbf{x}}$, we can find corresponding models with articulations $\mathbf{u}$. This can be done by intersecting the hyper-surfaces corresponding to all the models with the hyper-plane $\mathbf{x} = \bar{\mathbf{x}}$. For most points $\bar{\mathbf{x}}_i$, there will be relatively few corresponding models, because most models do not go through all points of the hyper-space even with articulation. Thus, the indexing space will be relatively sparse.

## Matching

Given a test range image, we want to determine which model in the dataset corresponds to the object in the image. Following the procedure in Section 10, we choose a particular value of the scale parameter and compute the invariant coordinates $\bar{\mathbf{x}}_i$ of data points taken from a uniform grid on the range image. We seek to determine the model in the dataset that has an articulation whose invariant spatial coordinates closely match these $\bar{\mathbf{x}}_i$'s.
Starting with the invariant coordinates of one data point $\bar{\mathbf{x}}_1$, we intersect all the hyper-surfaces with the hyper-plane $\mathbf{x} = \hat{\mathbf{x}}_1$ where $\hat{\mathbf{x}}$ is a discretization of $\mathbf{x}$ at the discretization level of the hyper-surface. Note that in order to do so, we need to extract a true representation of the hyper-surface from its compact form only in the vicinity of the hyper-plane $\mathbf{x} = \hat{\mathbf{x}}_1$ and not the entire space covered by the surface. Each individual hyper-surface will contribute a discrete subspace $\mathbf{u}_k(\hat{\mathbf{x}}_1)$ that will represent all the values of the articulation parameters that have spatial coordinates $\mathbf{x} = \bar{\mathbf{x}}_1$ in their corresponding subspaces. In other words, if there is only one articulation parameter, $\mathbf{u}_k(\hat{\mathbf{x}}_1)$ will be a one-dimensional vector that contains a 1 in those cells that represent a value of the articulation parameter that has the specified spatial coordinates, and contains 0 everywhere else. We repeat the process for another image point with invariant coordinates $\bar{\mathbf{x}}_2$ to obtain a different set of subspaces $\mathbf{u}_k(\hat{\mathbf{x}}_2)$, with each hyper-surface contributing a discrete representation of all the articulation parameters that have invariant spatial coordinates $\mathbf{x} = \bar{\mathbf{x}}_2$. Continuing with the other test invariant points $\bar{\mathbf{x}}_i$, we can accumulate all the $\mathbf{u}_k(\hat{\mathbf{x}}_i)$ and aggregate them as

$$\mathbf{U}_k = \sum_i \mathbf{u}_k(\hat{\mathbf{x}}_i)$$

Note that in the case of one articulation parameter, each $\mathbf{U}_k$ will be a one-dimensional vector, and if hyper-surface $f_j$ is derived from the corresponding model, then $\mathbf{U}_j$ will have a prominent peak at the cell that corresponds to the value of the articulation parameter present in the test range image. Consequently, we determine the peaks in each $\mathbf{U}_k$, and the models corresponding to the hyper-surfaces with the highest peaks will be candidate matches for possible further verification.

## Characteristics

The compact representation of the hyper-surface should have the following characteristics:

1. Since the invariant spatial coordinates for neighboring articulation parameter values are highly correlated, the representation should achieve compression by taking advantage of this smoothness. Moreover, if the degree of compression required is increased, then the size of the representation

should be reduced in such a manner that the reconstruction of the true representation does not degrade substantially.

2. Since the intersection of the hyper-surface with the hyper-plane $\mathbf{x} = \hat{\mathbf{x}}_1$ corresponds to extracting the true representation of the surface in the range $(\hat{\mathbf{x}}_1, \mathbf{u}_{min})$ to $(\hat{\mathbf{x}}_1, \mathbf{u}_{max})$, where $(\mathbf{u}_{min}, \mathbf{u}_{max})$ denote the extrema of values of the articulation parameters, it should be possible to extract this by decompressing the surface only in the vicinity of this range, and this should not require the decompression of the remaining portion of the surface.

It will be shown that using discrete wavelets to compress the surface allows us to accomplish both these objectives. Additionally, since wavelet compression usually involves more than one resolution level, this representation also provides the following benefit:

1. It is possible to generate the true representation of the surface in the desired range by levels. Each additional level provides a higher degree of resolution to the reconstruction. Therefore, the lower-resolution reconstructions can be used to discard hyper-surfaces that show accumulator peaks much smaller than the others.

# 14

# Model Dataset

This dataset consists of hyper-surfaces corresponding to different models. A hyper-surface for a model corresponds to a particular viewpoint and incorporates all possible articulations of the model components. Because the hyper-surface is created from invariant characteristics of points, it is reasonably view-invariant and thus the number of different viewpoints that need to be considered for a single model will be rather small. In fact, two different viewpoints need to be considered only when the components of the model visible in the two range images differ considerably.

## 14.1 Overview

The creation of the model dataset consists of the following steps:

1. For each model, determine the number of distinct viewpoints from which hyper-surfaces will be derived. Since two distinct viewpoints have to be considered only when the visible model components change considerably, the number of viewpoints will differ from model to model. Currently, no attempt has been made to determine the minimum number of such viewpoints that will be required for each model since the dataset for the experiments corresponded to hyper-surfaces of the models from a single viewpoint.

2. A hyper-surface is constructed from invariant characteristics derived from neighborhoods of fixed size and is thus a function of the neighborhood radius. We need to define a set of radius values at which the hyper-surfaces will be generated. Note that these radius values will be used to generate hyper-surfaces for all the models. While it is possible to select a single radius for a model, it is prudent to select a few contiguous discrete values and accumulate the evidence provided by the hyper-surfaces for each value. Moreover, in order to make the recognition process less susceptible to occlusion, these radius values should be chosen as small as possible without making them so small that the neighborhoods do not provide usable invariant characteristics.

3. For each model, create hyper-surfaces for those radius values that are valid for the model. A radius value becomes invalid for a model if the corresponding neighborhood encompasses the whole model. For practical purposes, these implicit surfaces are created in the form of discrete binary functions. They are then decomposed using wavelets of an arbitrarily chosen family since no attempt has been made to customize the wavelets. Compression is achieved by discarding coefficients in the decomposition that fall below a threshold. Since the resulting representation will be rather sparse, the compressed decomposition is stored using standard sparse matrix methods.

In the following sections, we describe how a single hyper-surface is created, how it is decomposed using wavelets, and how this decomposition is compressed and stored.

## 14.2 Hyper-surface creation

A hyper-surface for a model is created from range images of that model derived from a fixed viewpoint and corresponding to all possible articulations of the components of the model. It is the representation of the invariant characteristics of regularly spaced points on all such range images, with these characteristics being derived from a fixed neighborhood around each point. If we denote the invariant coordinates of a point on a range image corresponding to a particular value $r$ of the neighborhood radius by $\mathbf{x}_r$, then these coordinates will be functions of the values of the articulation parameters $\mathbf{u}$ corresponding to the range image, i.e. $\mathbf{x}_r(\mathbf{u})$. Consequently, the relationship between the invariant coordinates of all the points and the articulation parameters can be expressed in terms of the implicit hyper-surface $f_r(\mathbf{x}, \mathbf{u}) = 0$. To construct such a surface we vary the values of the articulation parameters $\mathbf{u}$, and for each value of $\mathbf{u}$ we determine all the corresponding invariant coordinates $\mathbf{x}$. The methods used to determine these coordinates are identical to those described in Part 1.

However, in order to represent this surface using discrete wavelets, we need to divide the domain of the function into discrete equal intervals, so that the implicit surface is effectively reduced to a binary function over the discrete domain. In other words, if $(\mathbf{x}, \mathbf{u})$ is a point on the original surface and $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$ is the index of the point in the discrete domain, then $\hat{f}_r(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = 1$.

The steps involved in the creation of the hyper-surface corresponding to a single viewpoint and a single radius value can be summarized as follows:

1. Divide the range of acceptable values of each articulation parameter into discrete intervals and choose the mid-point of each interval as the value corresponding to the interval. For all possible discrete values of the articulation parameters, generate range images of the model from the chosen viewpoint.

2. For each range image, generate a set of invariant coordinates as follows:

   (a) Select points on the range image using a regularly spaced grid. As discussed in Section 10.4, the resolution of this grid is not very significant as long as it is reasonably dense.

   (b) For each such point, fit a quadratic surface to its neighborhood using the chosen radius and derive a 3D canonical frame from the characteristics of this surface. The coordinates of the point in this canonical frame constitute its invariant coordinates. A detailed description of this process can be found in Sections 9.3.

3. Select a discretization level for each dimension of the hyper-surface in such a manner that a unit length along each geometric dimension represents approximately the same 3D length. Moreover, the level of the discretization along the geometric dimensions should be about half of the grid resolution chosen while selecting points on the range image in order to prevent gaps in the resulting surface. The discretization level along the articulation dimensions is the same as the discretization used while generating the range images. The hyper-surface is initialized as an empty multi-dimensional sparse matrix with the appropriate dimensions.

4. Map the generated invariant coordinates together with the corresponding values of the articulation parameters into the discrete domain of the hyper-surface and set the corresponding matrix elements

to 1. Note that while this introduces some discretization errors in the surface representation, the recognition process proves to be quite tolerant to them.

## 14.3 Hyper-surface representation

The binary multi-dimensional surface is decomposed using discrete wavelets belonging to an arbitrarily chosen family. No attempt was made to generate customized wavelets or compare representations with wavelets of different families. Note that while the coefficients in the decomposition will no longer be binary, the coefficient matrices will still be sparse.

Compression of the decomposition consists of discarding coefficients whose magnitudes are below a specified threshold. It was found experimentally that setting approximation coefficients to zero distorted the surface much more than discarding detail coefficients. Thus, in the current implementation, only detail coefficients are discarded while performing the compression. The compressed coefficient matrices are then stored on disk in the form of sparse matrices.

# 15

# Matching

In this section we describe how invariant characteristics derived from a test range image are compared against a hyper-surface and how a measure of goodness of the match is obtained. Note that in this process, we seek to determine not only which model corresponds to the object in the test image, but also the unknown values of the articulation parameters of the object.

## 15.1 Overview

The steps involved in the matching process are:

1. Attempt to segment out as much of the object as possible from the surrounding terrain using planarity information (see Section 10 for details).
2. Sample the test range image using a uniform grid and obtain a set of data points whose invariant characteristics will be considered. The resolution of the grid can be quite sparse since each grid point provides an independent piece of evidence during the matching process, though the reliability of the final match does depend on the number of points considered.
3. For each neighborhood radius in the pre-defined set, do the following:
   (a) Select the hyper-surfaces that correspond to this radius value.
   (b) Determine the invariant coordinates for each of the grid points as follows (see Section 9.3 for details):
      - fit a quadratic surface using a neighborhood around the grid point with the given radius value,
      - determine a 3D canonical frame using the characteristics of this surface,
      - map the grid point to the canonical frame to obtain its invariant coordinates.

   (c) Compare the set of invariant coordinates so obtained against each of the hyper-surfaces selected and obtain measures of the goodness of the match. These measures not only indicate the model involved in the match but also provide estimates of the values of the articulation parameters in the object present in the test image.
4. Terminate the search procedure when two consecutive radius values provide reasonably good measures for the same model and the same articulation value estimates.

## 15.2 Hyper-surface matching

Given a set of invariant points obtained from the test range image and a hyper-surface, both of which correspond to the same neighborhood radius value, we seek to determine how well the two match. This is achieved by determining, for each invariant point, the set of points in the surface that have the same geometric coordinates (though different articulation values), and recording the articulation values of the points in this set. We then determine the articulation values that appear most frequently in all such sets of articulation values. If the hyper-surface corresponds to the correct model, then it is assumed that the articulation value that appears most frequently will be close to the true articulation value of the object in the image.

The steps involved in this comparison process are:

1. For each invariant point that lies within the extents of the geometric dimensions of the hyper-surface, do the following:

   (a) Obtain the index $\hat{\mathbf{x}}$ of the point in the discrete domain of the surface.

   (b) Reconstruct the surface in the range $(\hat{\mathbf{x}}_1, \hat{\mathbf{u}}_{min} \cdots \hat{\mathbf{u}}_{max})$, where $\hat{\mathbf{u}}_{min}$ and $\hat{\mathbf{u}}_{max}$ are the minimum and maximum indices of the articulation dimensions. Call this reconstructed portion an *accumulator* for this invariant point. Since the accumulator will not be binary because of the errors introduced during compression, we threshold its elements to obtain binary values. For instance, if there is only one articulation parameter, the reconstructed portion will be a 1D vector; if there are two articulation parameters, the portion will be a 2D matrix; and so on.

2. Modify the individual accumulators using the iterative scheme described in Section 15.3.

3. Sum the accumulators for all the invariant points and normalize the result by dividing by the number of points. It is assumed that the final accumulator will have a peak at an articulation value that is close to the true one.

4. Identify the peaks in the final accumulator by determining zero crossings in the first differences along each dimension, and retain the magnitude and articulation values corresponding to the most prominent peaks.

## 15.3 Cooperative improvement

The accumulators obtained from the individual grid points are modified using a cooperative scheme similar to the relaxation labeling approach used in region growing. The basic idea is that accumulators from neighboring grid points should contribute the same articulation values. In other words, while considering the accumulator obtained from a particular grid point, we identify the positions where this accumulator is non-zero and enhance those positions at which the accumulators from the neighboring grid points are also non-zero. Similarly, we depress those accumulator positions that do not have any support from the neighboring accumulators. However, because of the discrete nature of the hyper-surface, neighboring accumulators may not support exactly the same positions. Consequently, we need to smear out the accumulators by convolving them with a smoothing filter.

Let $A_c$ be the accumulator from the current grid point and $A_n$ an accumulator from one of its neighbors. If $s$ represents a smoothing filter and $A^s = s \otimes A$ the smoothed accumulator, then the modified accumulator for the current grid point $\hat{A}_c$ is given by

$$\bar{A}_c \quad = \quad w_c A_c + w_+ \sum_n ((A^s_c > \epsilon_1) \circ A^s_n) - w_- \sum_n ((A^s_c \leq \epsilon_1) \circ A^s_n) \qquad (72)$$

$$\hat{A}_c = \frac{(\bar{A}_c > \epsilon_2) \circ \bar{A}_c}{\max(1, \max(\bar{A}_c))} \tag{73}$$

where $\epsilon_1$ and $\epsilon_2$ are positive thresholds identifying non-zero values, $A > \epsilon$ and $A \leq \epsilon$ are the binary results of the corresponding thresholding operations, $\circ$ denotes element-wise multiplication, and $w_c$, $w_+$ and $w_-$ are appropriate weights. The summation is carried out over a neighborhood of fixed size, and $w_+$ and $w_-$ are chosen to be inversely proportional to the number of neighbors considered. The operations in Equation 73 ensure that $\hat{A}_c$ has values between 0 and 1.

Thus, the cooperative scheme consists of modifying the individual accumulators using a neighborhood of fixed size for a small number of iterations. These iterative improvements will sharpen the peaks in the final accumulator while small variations will be smoothed out.

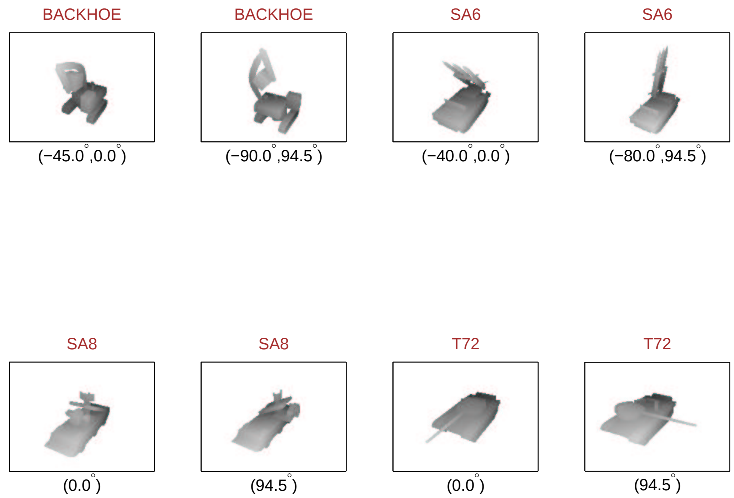


Figure 32: Sample reference range images of the models

# 16

# Results

Experiments were conducted to test this matching scheme with synthetic range data using four models. Each of these models had one or two articulated joints. In order to validate the ideas underlying the matching scheme, we seek to not only match the object in a test range image to the reference range image of the correct model where the viewpoint could be considerably different, but also estimate the values of the articulation parameters of the object in the test image. Some of the results in this section have been reported in [Reference 23].

## 16.1 Model dataset

The model dataset consists of four real-life models. Reference range images are generated for all possible articulations of the components of each model from a single viewpoint. Some of the range images used are shown in Figure 32. The specifications of these range images are listed in Table 3.

Invariant characteristics are extracted from points sampled from each of these range images using a reasonably dense grid. For each grid point, these characteristics are obtained for three different neighborhood radii corresponding to three different values of the scale parameter. For each model and for each scale value, the invariant points of all the reference range images of a model are mapped to a discrete hyper-surface with a discretization level chosen arbitrarily, subject to the constraints described in Section 14.2. The specifications for the surfaces of all the models at an intermediate scale value are shown in Table 4. Note that the resulting multi-dimensional matrices are relatively sparse.

These matrices are then decomposed using the Daubechies wavelet function of order 3. The detail

| Model | Image Size (width,height) | Viewpoint (az,el) | Articulations | # of ref. images |
|---|---|---|---|---|
| BACKHOE | (400,300) | $(30°, 30°)$ | Art. 1: 5 values b/w $[-90°, 0°]$<br>Art. 2: 20 values b/w $[0°, 360°]$ | 100 |
| SA6 | (400,300) | $(30°, 30°)$ | Art. 1: 5 values b/w $[-80°, 0°]$<br>Art. 2: 20 values b/w $[0°, 360°]$ | 100 |
| SA8 | (400,300) | $(30°, 30°)$ | Art. 1: 20 values b/w $[0°, 360°]$ | 20 |
| T72 | (400,300) | $(30°, 30°)$ | Art. 1: 20 values b/w $[0°, 360°]$ | 20 |

Table 3: Model range image specifications

| Model | Surface Dims. [Geom.] × [Art.] | # of non-zero coeff. (before thresholding) | # of non-zero coeff. (after thresholding) | % of coeff. retained |
|---|---|---|---|---|
| BACKHOE | [20,21,52] × [5,20] | 33,525 | 8,312 | 24.79% |
| SA6 | [20,25,60] × [5,20] | 37,828 | 8,921 | 23.58% |
| SA8 | [20,23,35] × [20] | 15,388 | 3,964 | 25.76% |
| T72 | [20,33,32] × [20] | 15,158 | 4,212 | 27.78% |

Table 4: Characteristics of the compressed hyper surfaces

coefficients of the resultant decomposition are thresholded, thus reducing the number of non-zero coefficients. The percentages of coefficients retained in the decomposition following the thresholding are shown in Table 4. The results presented in this section use these compressed surface representations for matching.

## 16.2 Our Test images

The test images include two views of each object in the model dataset. The first view of each object corresponds to a viewpoint that is the same as the viewpoint from which the reference range data was obtained, while the second view corresponds to a relatively disparate viewpoint. In both views, the articulation of the object is unknown and has to be estimated. The specifications for the test images are listed in Table 5.

Matching proceeds by sampling the test range images using a relatively coarse grid and determining invariant characteristics for each grid point at each scale value. Here, we report the results obtained at the

| Model | View | Image Size (width,height) | Viewpoint (az,el) | Articulations |
|---|---|---|---|---|
| BACKHOE | View 1 | (200,150) | $(30°, 30°)$ | $(-45°, 150°)$ |
| | View 2 | (200,150) | $(45°, 30°)$ | $(-65°, 250°)$ |
| SA6 | View 1 | (200,150) | $(30°, 30°)$ | $(-45°, 150°)$ |
| | View 2 | (200,150) | $(45°, 30°)$ | $(-65°, 250°)$ |
| SA8 | View 1 | (200,150) | $(30°, 30°)$ | $(150°)$ |
| | View 2 | (200,150) | $(45°, 30°)$ | $(80°)$ |
| T72 | View 1 | (200,150) | $(30°, 30°)$ | $(150°)$ |
| | View 2 | (200,150) | $(45°, 30°)$ | $(250°)$ |

Table 5: Specifications for the test range images

scale level corresponding to that of the surfaces specified in Table 4. The invariant coordinates of each grid point are used to reconstruct a portion of each surface, resulting in either an 1D or 2D accumulator. Each such accumulator is thresholded to obtain binary values and iteratively modified using a $5 \times 5$ neighborhood through 5 iterations. The smoothing filter used is a normalized version of the [1 2 1] filter. These modified accumulators are added to form the final accumulator, which is normalized by dividing by the number of accumulators. The results are presented in figures of two types:

1. *Accumulator Figures for the correct model* (Figures 33, 35, 37, 39, 41, 43, 45 and 47): These figures show the final accumulators obtained from the hyper-surfaces corresponding to the correct models. The grid points that contributed to the correct articulation values and those that did not are also indicated in each figure. Observe that in test images where the viewpoint is the same as the viewpoint of the model, the peaks obtained are very distinct, while in images where the viewpoint is relatively different the peaks degrade to some extent. However, in all cases, the correct articulation values are relatively close to the peaks in the accumulators.

2. *Accumulator Figures for all models* (Figures 34, 36, 38, 40, 42, 44, 46 and 48): These figures show the final accumulators obtained from all the hyper-surfaces for each test image. Note that the accumulator corresponding to the correct model has peak values that are more prominent than any peaks in the other accumulators, many of which contain no significant peaks at all. This holds true even when the viewpoint of the test image is quite different than the viewpoint from which the model range data is obtained. Thus, we are able to correctly identify the object in each test image and estimate its articulation values fairly accurately, though it must be pointed out that the number of models considered is quite small.

### 16.3 IRMA Test Images

In addition to the test images described above, we have also tested our method on a collection of range images provided by the Air Force. These images were generated using the IRMA simulation system which includes many realistic effects such as atmospheric disturbances.
A typical IRMA image has a target such as a tank lying on a hilly ground and contains non-target objects such as trees. Our method has succeeded in identifying the targets and rejecting the non-targets, subject to the following constraints:

- *Sphere radius:* The scale on which we are working, namely the radius of the spheres, is greater than about a quarter of the size of the object itself.

- Ground segmentation: The ground needs to be separated from the object prior to recognition. Being flat, the ground tends to produce "flat" conics, namely ones with long axes. Such axes tend to be obtained less reliably. The problem can be easily solved by finding and isolating flat parts of the image.

*Note on efficiency:* The most time consuming part of the method is the integration over each sphere, since an integral depends on calculating two $10 \times 10$ matrices at each point of the sphere. There is a sphere at each grid point, which means a large number of integrations. A way to reduce the integration time is to take advantage of the overlap between spheres at nearby points. Because of this overlap, many points of the object are visited many times, with the same lengthy calculation repeated there every time. This can be avoided by "tiling" the image. That is, we divide the image into relatively small rectangular tiles (in $x, y$ coordinates), and integrate over each tile. This is done only once, before dealing with the spheres. The integration domain of each sphere is made up (approximately) of a set of such tiles. Instead of integrating over a sphere we now simply sum up the integral values of the tiles that it contains. In this way, we have speeded up the execution time by a factor of about 50, with only a small degradation in accuracy.

# 17

# Conclusions

In this project we have implemented a novel method of recognizing articulated objects in range images and evaluated some of the key elements of this method.

1. *Segmentation into spheres.* We used a representation of the object based on dividing the image into spheres centered around grid points. Each grid point was assigned some numerical values derived from the object part contained within the sphere surrounding it. We have seen that this method enables to avoid an overly local representation that depends on small neighborhoods around the grid points. Such local methods are quite sensitive to noise and other local distortions. At the same time we were able to recognize objects given values on only some of the grid points, unlike methods that require the whole object.

2. *Invariant representation.* The representation of the object is Euclidean invariant. That is, the grid values are constructed in a way that makes them invariant to rotation and translation. This was done by fitting a quadric surface to the the object part and finding the invariants of the quadric. Several problems had to be overcome in this process as described earlier, but we obtained a quite stable and useful invariant representation that eliminated the need to look for the correct viewpoint.

3. *Cooperative algorithm.* Neighboring grid points should have similar values. We used this fact to as a basis for a cooperative algorithm, in which neighbors with similar values strengthen each other while a point which stands out from its neighborhood is weakened. This greatly improved the reliability in identifying the object as well as in finding its articulation parameters and its pose.

4. *Implicit surface representation.* We have used a multi-dimensional surface to represent the models in a hyper-space parametrized by both the spatial coordinates and the articulation parameters. The spatial coordinates are the invariant descriptors, or grid values. Given the invariants from the given object, we represented them as a hyper-plane in this hyper-surface, spanned by the unknown articulation parameters. The intersections of this plane with the surface gives us the articulation parameters. Combined with the cooperative algorithm mentioned above we obtained very reliable results.

In future work we intend to refine the process of finding invariants of the object parts, using weighted vectors derived from both quadric and plane fitting.

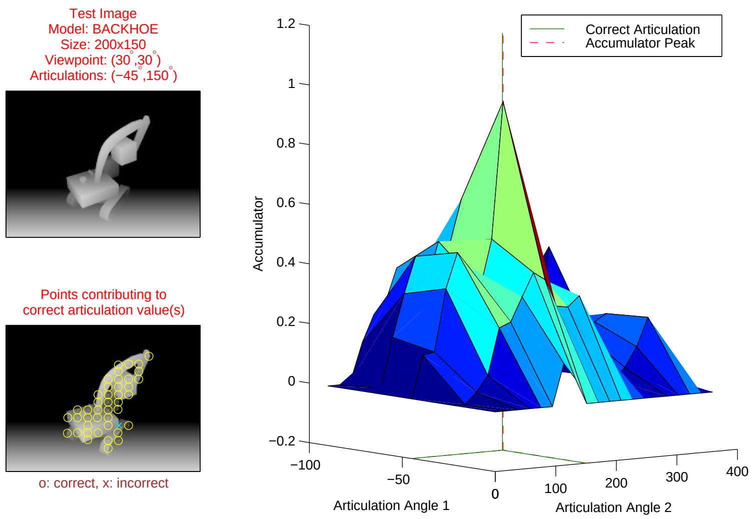


Figure 33: Model BACKHOE, View 1: Accumulator from the correct model surface

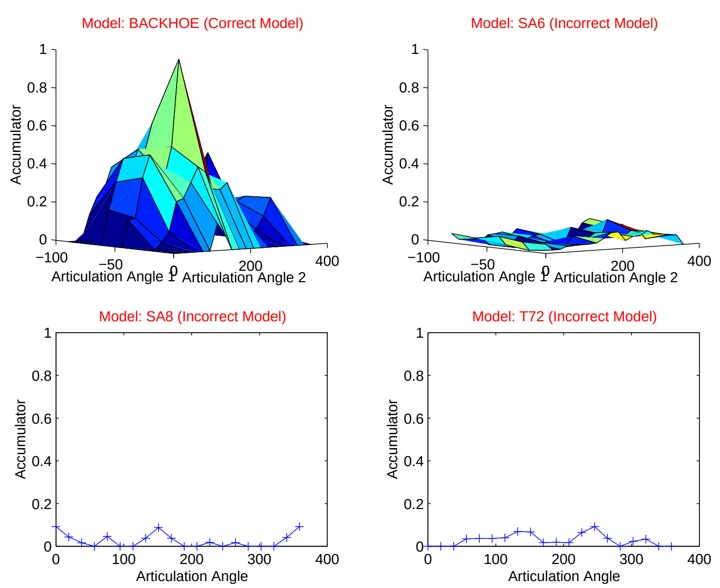


Figure 34: Model BACKHOE, View 1: Accumulators from all model surfaces

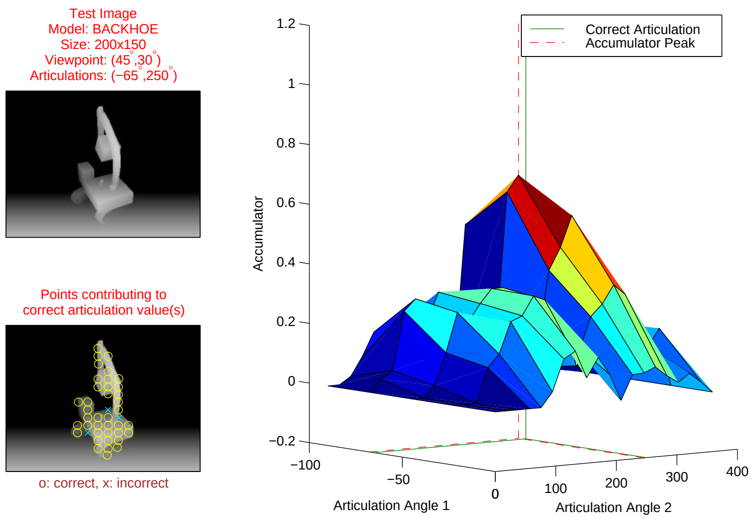


Figure 35: Model BACKHOE, View 2: Accumulator from the correct model surface

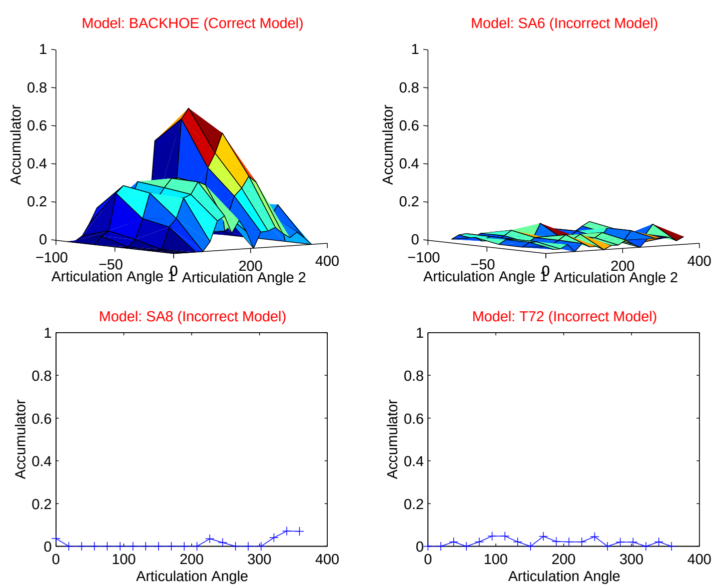


Figure 36: Model BACKHOE, View 2: Accumulators from all model surfaces

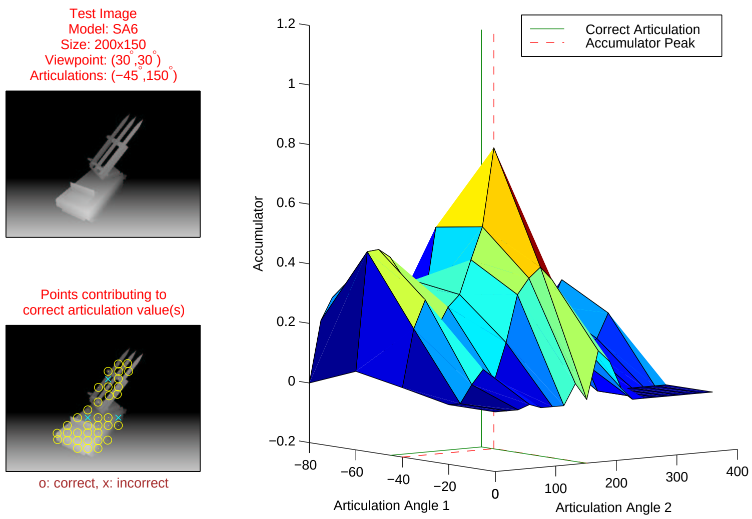


Figure 37: Model SA6, View 1: Accumulator from the correct model surface

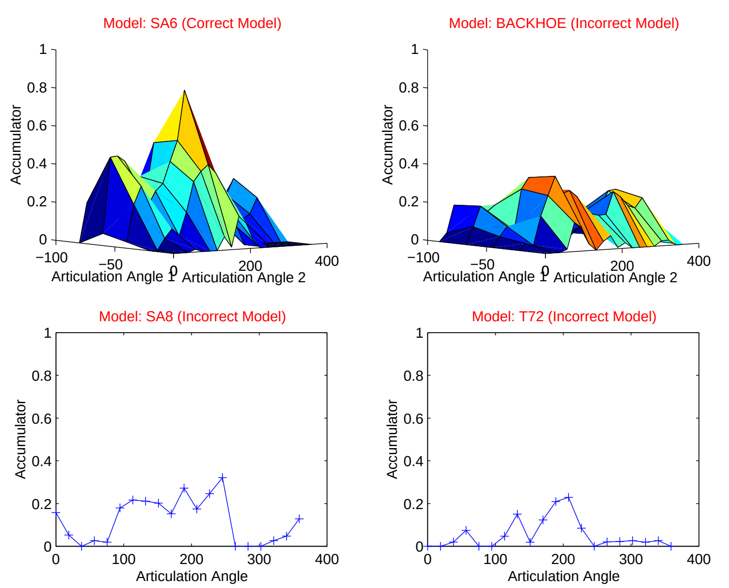


Figure 38: Model SA6, View 1: Accumulators from all model surfaces

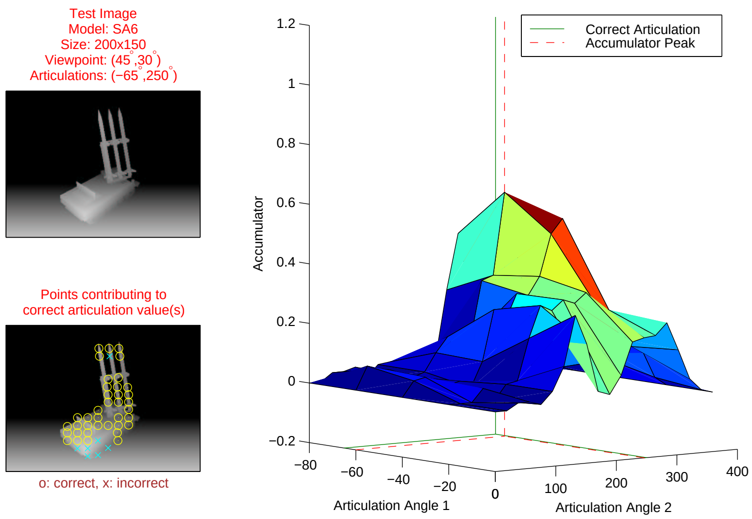


Figure 39: Model SA6, View 2: Accumulator from the correct model surface

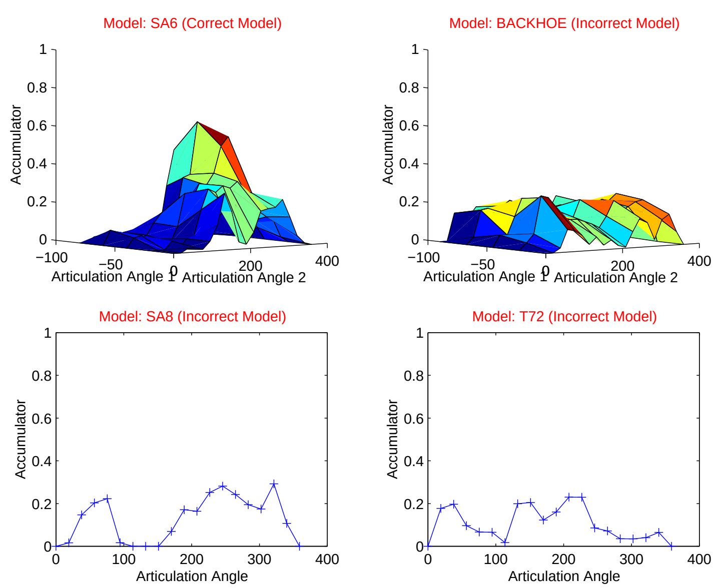


Figure 40: Model SA6, View 2: Accumulators from all model surfaces

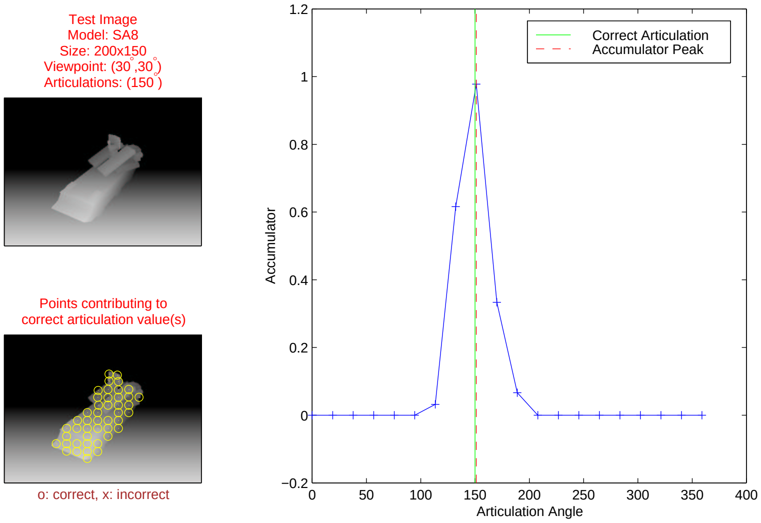


Figure 41: Model SA8, View 1: Accumulator from the correct model surface

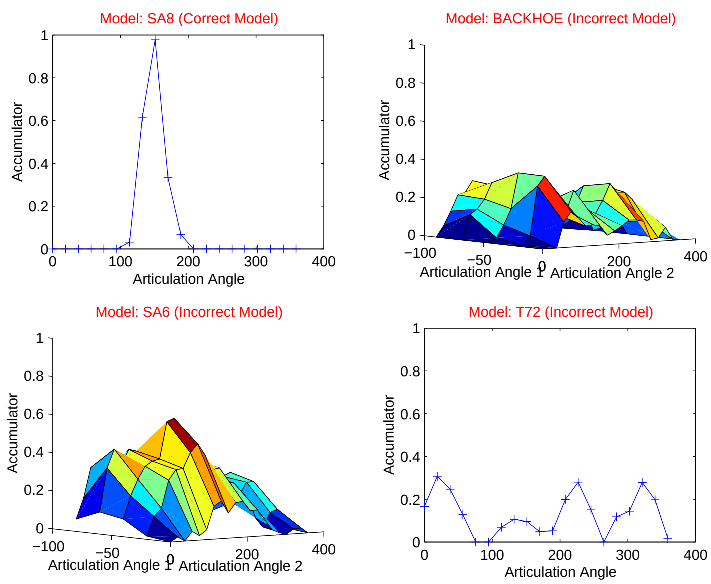


Figure 42: Model SA8, View 1: Accumulators from all model surfaces

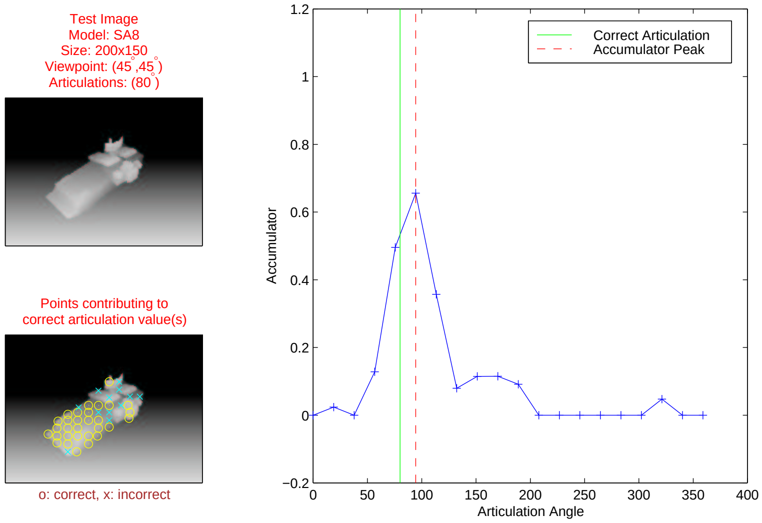


Figure 43: Model SA8, View 2: Accumulator from the correct model surface

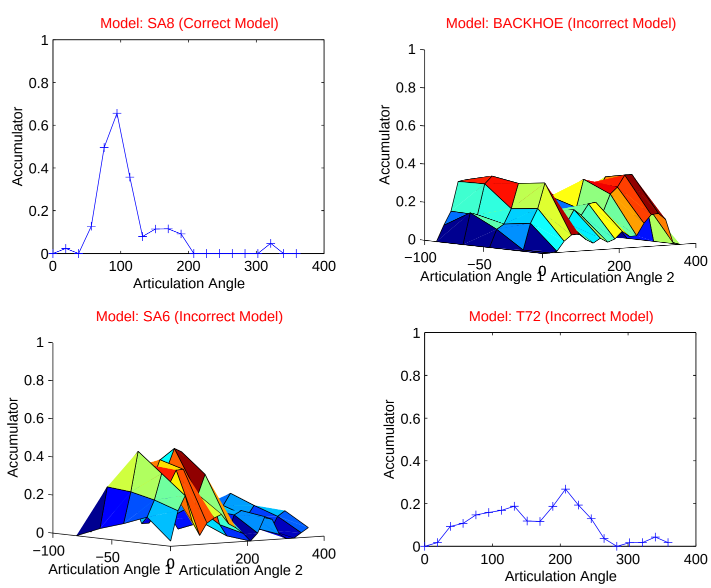


Figure 44: Model SA8, View 2: Accumulators from all model surfaces

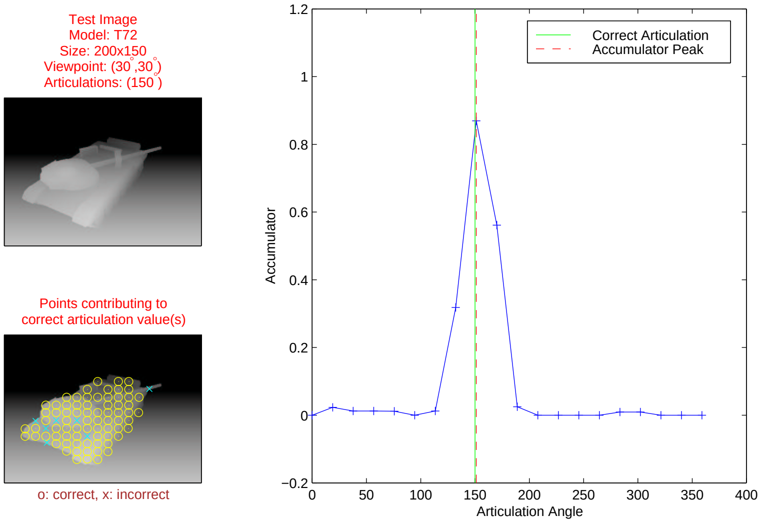


Figure 45: Model T72, View 1: Accumulator from the correct model surface

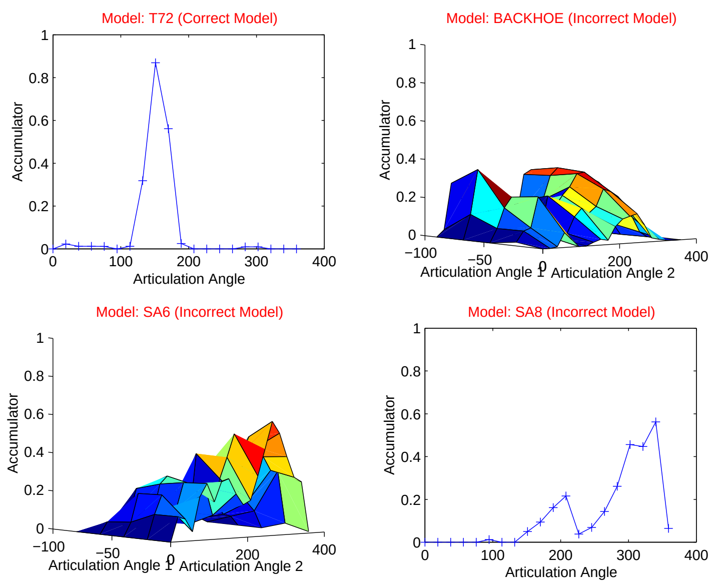


Figure 46: Model T72, View 1: Accumulators from all model surfaces

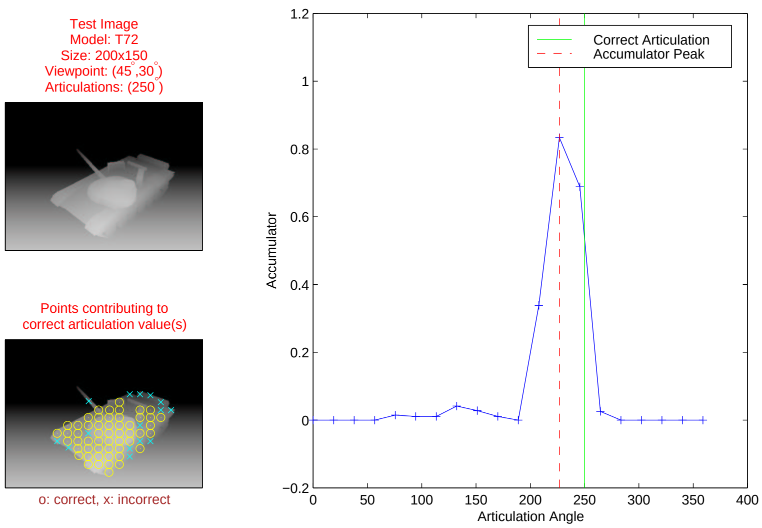


Figure 47: Model T72, View 2: Accumulator from the correct model surface

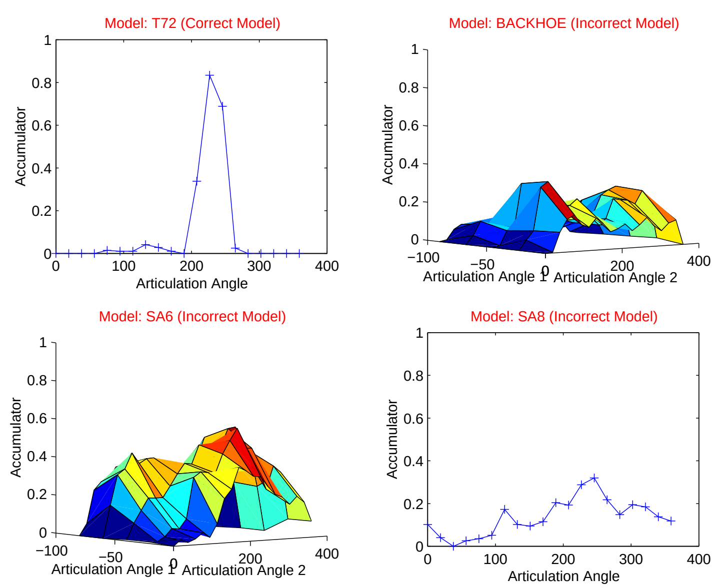


Figure 48: Model T72, View 2: Accumulators from all model surfaces

# 18

# References


[1] F. Arman and J. K. Aggarwal. Model-based object recognition in dense-range images – A review. *ACM Computing Surveys*, 25:5–43, 1993.

[2] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[3] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:167–192, 1988.

[4] T. O. Binford. Visual perception by computer. In *IEEE Conference on Systems and Control*, Miami, FL, 1971.

[5] T. J. Fan, G. Medioni, and R. Nevatia. Segmented descriptions of 3-D surfaces. *IEEE International Journal on Robotics and Automation*, 3:527–538, 1987.

[6] G. D. Godin and M. D. Levine. Structured edge map of curved objects in a range image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 276–281, San Diego, CA, 1989.

[7] A. Hauck, S. Lanser, and C. Zierl. Hierarchical recognition of articulated objects from single perspective views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 870–876, San Juan, Puerto Rico, 1997.

[8] R. Hoffman and R. C. Jain. Segmentation and classification of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:608–620, 1987.

[9] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[10] C. L. Jackins and S. L. Tanimoto. Octrees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14:249–270, 1980.

[11] J. J. Koenderink and A. J. Van Doorn. The internal representation of solid shape with respect to vision. *Biological Cyberbetics*, 32:211–216, 1979.

[12] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:441–450, 1991.

[13] F. Mokhtarian. Multi-scale description of space curves and three-dimensional objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 298–303, Ann Arbor, MI, 1988.

[14] C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis*, 10, 1973.

[15] J. Ponce and M. Brady. Toward a surface primal sketch. In T. Kanade, editor, *Three-Dimensional Machine Vision*, pages 195–240. Kluwer Academic Publishers, Boston, MA, 1987.

[16] M. Sallam, J. Stewman, and K. Bowyer. Computing the visual potential of an articulated assembly of parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 636–643, 1990.

[17] F. Solina and R. Bajcsy. Recovery of parametric model from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:131–147, 1990.

[18] T. Sripradisvarakul and R. Jain. Generating aspect graphs for curved objects. In *Proceedings of the IEEE Workshop on the Interpretation of 3-D Scenes*, pages 109–115, 1989.

[19] M. D. Swanson and A. H. Tewfik. A binary wavelet decomposition of binary images. *IEEE Transactions on Image Processing*, 5:1637–1650, 1996.

[20] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. Technical Report 7, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1994.

[21] G. Taubin. Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1115–1138, 1991.

[22] I. Weiss. Projective Invariants of Shape. In *Proc. Computer Vision and Image Processing*, 291–297, Ann Arbor, 1988.

[23] I. Weiss and M. Ray. Recognizing articulated objects using invariance. In *Proceedings of the International Conference on Pattern Recognition*, 2000. To appear.

[24] I. Weiss and M. Ray. Model-Based Recognition of 3D Objects from Single Images. *IEEE T-PAMI*, 23(2):116–128, Feb. 2001.

[25] M. Welfare and K. Norris-Zachery. Characterization of articulated vehicles using LADAR seekers. In *Proceedings of the SPIE Conference on Laser Radar Sensors and Systems*, 1997.

[26] H. S. Yang. Range image analysis via quad-tree and pyramid structure based on surface curvature. In D. P. Casasent, editor, *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision*, pages 597–608. Cambridge, MA, 1988.